高等院校计算机教育系列教材



Linux 网络技术基础

马 军 张 航 吴 涛 编著

- 知识点新，突出实践教学，强化能力培养
- 理论知识+感性认识+动手实践，完美结合
- 内容简明扼要，突出知识要点
- 以实用为宗旨，实例丰富，用实例引导读者模仿学习



清华大学出版社

高等院校计算机教育系列教材

Linux 网络技术基础

马　军　张　航　吴　涛　编著

清华大学出版社
北　京

内 容 简 介

全书结合企业版 CentOS 5.5 操作系统，全面而详细地介绍 Linux 操作系统的使用以及各种服务器的搭建过程。全书从零开始深入透彻地讲解 Linux 系统的基础知识，同时还结合传统的 UNIX 操作系统讲解相关知识。通过本书读者可以学习到如何有效使用 Linux 系统，理解并掌握命令行功能、文件系统、用户和组、bash shell、文本编辑器、网络服务器的架设，以及图形应用方面的知识和技能。

全书共分为 11 章，第 1～2 章介绍 Linux 的基础知识和 CentOS 的安装过程，以及 Linux 图形桌面系统的使用技巧；第 3 章介绍 Linux 下的一些常用命令以及 Linux 中网络配置文件的使用；第 4～10 章以每章一个重点的形式介绍了 Linux 各种网络服务器的搭建方法和技巧，包括 DHCP 服务器、NFS 服务器、DNS 服务器、Samba 服务器、Web 服务器、FTP 服务器，以及 Mail 服务器中常用的各种软件及使用方法。第 11 章则全面系统地介绍了 Linux 服务器的安全技术。综观全书，既有宏观的指导，也有微观细节的介绍；既有生动的实例讲解，也有典型经验的分享。

本书由浅入深，循序渐进，适合刚接触 Linux 的初学者使用，也可作为高等院校相关专业本科生、研究生的教材。

图书在版编目(CIP)数据

Linux 网络技术基础/马军，张航，吴涛编著. --北京：清华大学出版社，2012
(高等院校计算机教育系列教材)
ISBN 978-7-302-29620-1

Ⅰ. ①L…　Ⅱ. ①马…　②张…　③吴…　Ⅲ. ①Linux 操作系统—高等学校—教材　Ⅳ. ①TP316.89

中国版本图书馆 CIP 数据核字(2012)第 176888 号

责任编辑：汤涌涛
封面设计：杨玉兰
责任校对：王　晖
责任印制：

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn, http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　　邮　　编：100084
社 总 机：010-62770175　　　邮　　购：010-62786544
投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn, 010-62791865
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张：**22　　**字　数：**559 千字
版　　次：2012 年 9 月第 1 版　　**印　次：**2012 年 9 月第 1 次印刷
印　　数：1～4000
定　　价：39.00 元

产品编号：

前　　言

为何编写本书

Linux 作为一种自由和开放源码的类 Unix 操作系统，自从其诞生以来就受到了众多开发人员和企业用户的追捧。其发展虽然源于 Unix 操作系统，但适用范围和影响已经远远超过 Unix 系统本身。其流行的主要原因是因为它具有许多诱人之处，包括但不限于以下特点：

- 完全免费。
- 完全兼容 POSIX 1.0 标准。
- 多用户、多任务。
- 良好的用户界面。
- 丰富的网络功能。
- 可靠的安全、稳定性能。
- 支持多种平台。

现在，采用 Linux 作为操作系统的超级电脑也越来越多。根据 2008 年 11 月的 TOP500 超级电脑列表，现时世上最快速的超级电脑是使用 Linux 作为其操作系统的。在列表的 500 套系统里，采用 Linux 作为操作系统的占了 439 套(即 87.8%)。

可以说，不管是在商用领域还是在民用领域，Linux 都展现出强劲的实力。而学习 Linux 首先需要学习 Linux 的系统管理，这也正是撰写本书的初衷。

本书内容特色

1. 内容新颖、知识全面

全书从 Linux 的基础知识开始，逐渐引导读者了解 Linux 操作系统的基础知识，详细讲述了 Linux 系统的基本操作命令之后，又详细介绍了各种 Linux 服务器的搭建方法，做到了知识的全覆盖。

全书每一章都目标明确、重点突出，开篇给出了本章的学习要点和主要内容，能够帮助读者全面了解学习重点，制订学习计划。

本书内容由浅入深，从基础知识讲起，对读者的专业知识没有要求，为了避免学习的枯燥性，全书采用图文并茂的形式，以提高读者学习的兴趣。

2. 层次分明，学习轻松

本书结合作者多年的 Linux 使用经验，借鉴了许多资深 Linux 系统管理员的经验教训，从基本操作到服务器架设，全面介绍了 Linux 系统管理的方方面面，其内容翔实、层次分明，为想要成为 Linux 系统管理员的读者提供了很好的参考。

3. 通俗易懂，针对性强

本书适合想要学习 Linux 使用的读者，同时也适合想要进一步掌握 Linux 管理技巧的读者。本书并没有运用过多的专业术语，而是采用通俗易懂的文字、清晰形象的图片，便于读者理解和阅读，从而帮助读者快速掌握相关的技能和技巧。

本书内容丰富全面，对基本概念的讲解非常细致，深入浅出。各种功能和命令的介绍，都配以大量的实例操作和详尽的解析。本书是初学者学习 Linux 不可多得的一本好书。

适用读者群

- Linux 操作系统的初学者。
- 热爱开源软件以及 Linux 使用的用户。
- 可作为大中专院校或者社会培训的教材。
- 适用于 Linux 服务器管理员的参考用书。

目　　录

第 1 章

Linux 简介

Linux 操作系统是一种免费、开源的 Unix 类操作系统，它继承了 Unix 功能强大、性能稳定等的特点，具有良好的硬件平台移植性，是现阶段服务器级操作系统的首选。在本章中，我们主要介绍有关 Linux 操作系统的基础知识，包括 Linux 的起源、Linux 的发展历史、Linux 的内核及桌面环境以及 Linux 的发行版本等。

1.1 Unix 发展历史

要了解 Linux 的发展历史，不得不先了解 Unix 操作系统的发展历史。

1.1.1 Unix 简介

Unix 操作系统是美国 AT&T 公司于 1971 年在 PDP-11 上运行的操作系统。具有多用户、多任务的特点，支持多种处理器架构，最早由肯•汤普逊(Kenneth Lane Thompson)、丹尼斯•里奇(Dennis MacAlistair Ritchie)和 Douglas McIlroy 于 1969 年在 AT&T 的贝尔实验室开发的。

1.1.2 Unix 发展历程

早期的 Unix 操作系统的源代码是可以免费获得的，但是当 AT&T 发布 Unix 7 的时候，它认识到了 Unix 的商业价值，于是在发布的版本许可证中，不再允许大学在开设的课程中讲授其源代码。随后，AT&T 基于版本 7 开发了 Unix System Ⅲ的第一个版本，这个版本成为了真正的商业版本，仅供出售。

在 Unix 发展走向商业化的过程中，为了对抗商业化所带来的种种限制和诸多问题，柏克利大学在 1978 年以 Unix 第六版为基础，开发了仍然开放源码的操作系统 1 BSD(First Berkeley Software Distribution)，与 AT&T 的 Unix 形成了分庭抗争的局面。但是，随着 USL(Unix Systems Laboratories，AT&T 的附属公司)状告 BSD 剽窃其源码，致使 BSD 操作系统的发展陷于停滞，也正是在此阶段，Linux 这一开源因而没有版权问题的操作系统才得以发展、壮大。

随着 1994 年 USL 与 BSD 的版权诉讼案了结，BSD Unix 才再一次走向了复兴的道路。随后，BSD 的发展也开始多元化，相继产生了多种可以满足不同需求的操作系统，如 FreeBSD、OpenBSD 和 NetBSD 等。

1.1.3 Unix 版本介绍

现阶段比较著名的 Unix 版本有：

- Solaris 是 SUN 公司研制的类 Unix 操作系统。目前最新版本为 Solaris 10。早期的 Solaris 是由 BSD Unix 发展而来。这是因为 SUN 公司的创始人之一，比尔•乔伊(Bill Joy)来自伯克利大学(U.C. Berkeley)。但是随着时间的推移，Solaris 在接口上正在逐渐向 System V 靠拢。目前，Solaris 仍旧属于私有软件。
- Mac OS 是苹果公司开发的专属操作系统。Mac OS X 于 2001 年首次在市场上推出，并从 2002 年起随麦金塔电脑开始发售。它是一套以 Unix 为基础的操作系统，包含两个主要的部分：核心名为 Darwin，是以 FreeBSD 源代码和 Mach 微核心为基础，由苹果公司和独立开发者社区协力开发；一个由苹果电脑开发，名为 Aqua 的专有版权的图形用户界面。

图 1-1 给出了在 Unix 发展历程中纷繁复杂的众多版本的基本关系。

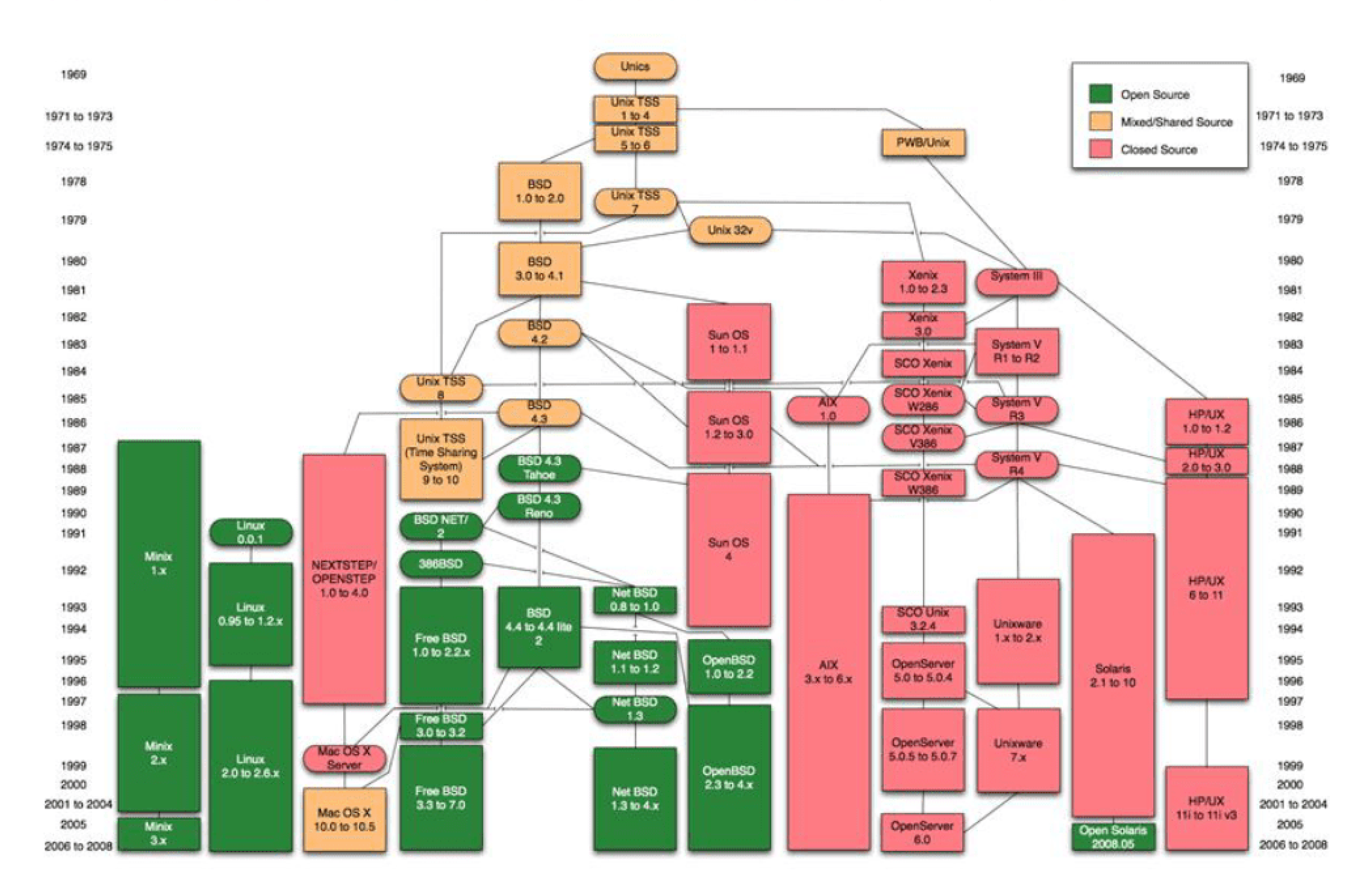


图 1-1　Unix 发展图谱

Unix 是第三次工业革命中计算机软件领域最具代表性的产物。在这近 40 年中，由 Unix 造成的影响是最有深远意义的。其遵循的原则一直都是 KISS(Keep It Simple, Stupid，尽量简单)。不能不说，AT&T 虽然发展了 Unix，但今天 Unix 的混乱局面也和 AT&T 有着直接关系。但反过来说，如果没有 AT&T 的反面教材，今天的 Linux 很有可能也不会出现。AT&T 究竟是限制了 Unix 的发展，还是以反面示例促进了 Unix 社区，已不好评说。今天，软件是商业化好还是开源好的争论还在继续，纵观这几十年来 Unix 的发展历史，Linux 划时代地出现，相信每个人心中会得出自己的结论。不管怎么样，Unix 的经历对计算机领域贡献的不单单是技术，它给我们提供了丰富而生动的教材。特别是 Unix 引发的哲学，让今天的我们依然受益匪浅。

1.2　Linux 发展历史

在了解 Linux 的发展过程之前，我们必须首先提及另一个操作系统——Minix，其名称来源于 Mini-Unix 的缩写。在上节中我们讲到，当 AT&T 发行 Unix 7 版本时，不再允许各大学在其课程中讲授 Unix 的源代码。为了改变这种局面，任职于荷兰阿姆斯特丹 Vrije 大学计算机科学系的 Andrew S. Tanenbaum 教授编写了一个与 Unix 完全兼容，而使用全新内核的操作系统，并命名为 Minix。Minix 系统充分继承了 Unix 的 KISS 原则，非常短小精悍，方便用于教学。但也正是因为这个原因，Minix 系统并不适合商业用途。而随着 Minix 用户的规模和使用途径的不断增多，迫切需要一种同样类似于 Unix，而又能够为大多数人服务，可以自由添加新特性的操作系统，正是在这种环境下，Linux 操作系

统诞生了。

1990 年，芬兰学生林纳斯·本纳第克特·托瓦兹(Linus Benedict Torvalds)开始编写一个类 Minix 的操作系统，并且在 comp.os.minix 新闻组中发布了这条消息，很快，互联网中众多高水平的程序员就加入了这个操作系统的开发过程。1991 年 10 月 5 日，Linus 正式发布了 Linux 0.0.2 版本，标志着 Linux 操作系统的诞生。之后，Linux 操作系统继续借助互联网的力量迅速壮大，越来越多的程序员加入到了这个完全开源并且自由扩散的操作系统的开发中，到 1993 年底，Linux 1.0 终于发布了，这已经是一个功能完备的操作系统了，其内核写得紧凑、高效，可以充分发挥硬件的性能。

Linux 最初是运行在 X86 硬件环境中的，但随着使用得越来越广泛，从 Linux 1.3 版本开始向其他硬件平台移植。现在 Linux 已经可以在几乎所有可见的主流硬件平台中运行，并且囊括了从低端到高端的所有应用。

从一开始，Linus 就决定 Linux 自由扩散并且遵循 GPL 原则，但是又不排斥商家的参与，于是在 Linux 上开发商业软件使 Linux 开始了新的飞跃。出现了很多 Linux 的发行版本，如大名鼎鼎的 RedHat 系列，Suse、Slackware、OpenLinux 等众多版本，而且，为了规避版权问题，很多公司还将其他 Unix 平台的软件移植到 Linux 上来，包括 IBM、Intel、Oracle、Sysbase、Corel、CA、Novell 等众多大牌厂商都宣布支持 Linux。众多商家的加入也弥补了自由软件的不足和发展的障碍，使 Linux 得以迅速普及。

提示： GPL(General Public Licence)是一种版权协议，它允许公众享有运行、复制软件的自由，发行传播软件的自由，获得软件源码的自由，改进软件并将自己做出的改进版本向社会发行传播的自由。

1.3 Linux 内核与桌面环境

内核是整个 Linux 系统的心脏，是运行程序和管理磁盘、打印机等硬件设备的核心程序，它接受用户的命令并执行。

图形操作系统中，一个桌面环境(Desktop Environment，有时称为桌面管理器)为计算机提供一个图形用户界面(GUI)。这个名称来自桌面比拟，对应于早期的文字命令行界面(CLI)。一个典型的桌面环境提供图标、视窗、工具栏、文件夹、壁纸以及像拖放这样的功能。整体而言，桌面环境在设计和功能上的特性，赋予了它与众不同的外观和感觉。

1.3.1 Linux 内核

总体来说，Linux 的内核主要包含以下功能：存储管理、中断异常与系统调用、进程与进程调度、文件系统、进程间通信、设备驱动、多处理器系统结构、系统引导与初始化等。

Linux 内核实现了很多重要的体系结构属性。在不同的层次上，内核被划分为多个子系统。Linux 也可以看作是一个整体，因为它会将所有这些基本服务都集成到内核中。这与微内核的体系结构不同，后者会提供一些基本的服务，例如，通信、I/O、内存和进程管理，更具体的服务都是插入到微内核层中的。

随着时间的流逝，Linux 内核在内存和 CPU 使用方面具有较高的效率，并且非常稳定。但是对于 Linux 来说，最为有趣的是在这种复杂的条件下，依然具有良好的可移植性。Linux 编译后可在大量处理器和具有不同体系结构约束和需求的平台上运行。Linux 可以在一个具有内存管理单元(MMU)的处理器上运行，也可以在那些不提供具有内存管理单元的处理器上运行。Linux 内核的 uClinux 移植提供了对非内存管理单元的支持。

1.3.2　Linux 内核版本

Linux 核心的开发和规范一直是由 Linux 社区控制着，版本也是唯一的。实际上，操作系统的内核版本指的是在 Linus 本人领导下的开发小组开发出的系统内核的版本号。自 1994 年 3 月 14 日发布了第一个正式版本 Linux 1.0 以来，每隔一段时间就有新的版本或其修订版公布。

Linux 内核版本发布的官方网站是 http://www.kernel.org，新版本的发布主要分为两种形式：一种是 Full Source 版本；另一种是 Patch 版本，即版本补丁。Full Source 版本体积比较大，一般是 tar.gz 或者.bz2 文件，两者分别是 gzip 和 bzip2 的压缩文件，使用时需要先解压缩；而 Patch 版本比较小，一般只有几十 KB 到几百 KB，但是 Patch 文件是针对特定版本的，用户需要找到对应的版本才能使用。

一般的，用户可以从 Linux 内核版本号来区分系统是 Linux 稳定版还是测试版。以版本 2.4.0 为例，2 代表主版本号，4 代表次版本号，0 代表改动较小的末版本号。在版本号中，序号的第二位为偶数的版本表明这是一个可以使用的稳定版本，如 2.2.5；而序号的第二位为奇数的版本一般有一些新的功能加入，是不稳定的测试版本，如 2.3.1。这样，稳定版本号来源于上一个测试版升级版本号，而一个稳定版本发展到完全成熟后就不再发展。

表 1-1 给出了自 Linux 出现以来的重要版本号及其更新内容。

表 1-1　Linux 内核版本号及发展历程

内核版本号	时　间	内核发展及更新内容
0.00	1991.2.4	两个进程分别显示 AAA 和 BBB
0.01	1991.9	第一个正式向外公布的 Linux 内核版本
0.02	1991.10.5	Linus Torvalds 将 0.02 内核版本发布到了 Minix 新闻组，很快就得到了反应，在很多热心支持者的帮助下，推出了 Linux 的第一个稳定工作版本
0.03	1991.10.5	
0.10	1991.10	Linux 0.10 版本发布，0.11 版本随后在 1991 年 12 月推出，当时它被发布在 Internet 上，供人们免费下载使用
0.11	1991.12.8	基本可以正常运行的内核版本
0.12	1992.1.15	主要加入对数学协处理器的软件模拟程序
0.95	1992.3.8	开始加入虚拟文件系统思想的内核版本
0.96	1992.5.12	开始加入网络支持和虚拟文件系统
0.97	1992.8.1	

续表

内核版本号	时　间	内核发展及更新内容
0.98	1992.9.29	
0.99	1992.12.13	
1.0	1994.3.14	Linux 1.0 版本内核发布，使用它的用户越来越多，而且 Linux 系统的核心开发队伍也建起来
1.2	1995.3.7	
2.0	1996.2.9	
2.2	1999.1.26	
2.4	2001.1.4	Linux 2.4.0 版本内核发布
2.6	2003.12.17	Linux 2.6 版本内核发布，与 2.4 内核版本相比，它在很多方面进行了改进，如支持多处理器配置和 64 位计算，它还支持实现高效率线和处理的本机 POSIX 线程库(NPTL)。实际上，性能、安全性和驱动程序的改进是整个 2.6.x 内核的关键
2.6.15	2006	Linux 2.6.15 版本内核发布。它对 IPv6 的支持在这个内核中有了很大的改进。PowerPC 用户现在有了一个用于 64 位和 32 位 PowerPC 的泛型树，它使这两种架构上的内核编辑成为可能
2.6.30	2009.6	改善了文件系统、加入了完整性检验补丁、TOMOYO Linux 安全模块、可靠的数据报套接字(Datagram Socket)协议支持、对象存储设备支持、FS-Cache 文件系统缓存层、nilfs 文件系统、线程中断处理支持等
2.6.32	2009.12	增添了虚拟化内存 de-duplicacion、重写了 writeback 代码、改进了 Btrfs 文件系统、添加了 ATI R600/R700 3D 和 KMS 支持 、CFQ 低传输延迟时间模式、perf timechart 工具、内存控制器支持 soft limits、支持 S+Core 架构、支持 Intel Moorestown 及其新的固件接口、支持运行时电源管理，以及新的驱动
2.6.34	2010.5	添加了 Ceph 和 LogFS 两个新的文件系统，其中前者为分布式的文件系统，后者是适用于 Flash 设备的文件系统。Linux Kernel 2.6.34 的其他特性包括新的 Vhost net、改进了 Btrfs 文件系统、对 Kprobes jump 进行了优化、新的 perf 功能、RCU lockdep、Generalized TTL Security Mechanism (RFC 5082)及 privateVLAN proxy arp (RFC 3069)支持、asynchronous 挂起恢复等
2.6.36	2010.10	Tilera 处理器架构支持、新的文件通知接口 fanotify、Intel 显卡上实现 KMS 和 KDB 的整合、并行管理工作队列、Intel i3/5 平台上内置显卡和 CPU 的智能电源管理、CIFS 文件系统本地缓存、改善虚拟内存的层级结构，提升桌面操作响应速度、改善虚拟内存溢出终结器的算法、整合了 AppArmor 安全模型(注：与 SELinux 基于文件的标注不同，AppArmor 是基于路径的)

1.3.3　桌面环境

在 Linux 操作系统中，桌面环境不仅仅是一个简单的窗口管理器，而是指一套完整的桌面应用程序套件。桌面环境提供一个功能强大、界面友好的交互环境，往往还包括文件管理器、图形处理软件和文字、表格处理软件等。

1.3.4　常用桌面环境介绍

现今主流的桌面环境有 KDE、GNOME、Xfce、LXDE 等，除此之外还有 Ambient、EDE、IRIX Interactive Desktop、Mezzo、Sugar、CDE 等。

1. KDE

KDE(Kool Desktop Environment)项目始建于 1996 年 10 月，相对于 GNOME 还要早一些。KDE 项目是由图形排版工具 Lyx 的开发者、一位名为 Matthias Ettrich 的德国人发起的，目的是为了满足普通用户也能够通过简单易用的桌面来管理 Unix 工作站上的各种应用软件以及完成各种任务。

2. GNOME

GNOME，即 GNU 网络对象模型环境(The GNU Network Object Model Environment)，GNU 计划的一部分，开放源码运动的一个重要组成部分，是一种让使用者容易操作和设定电脑环境的工具。

目标是基于自由软件，为 Unix 或者类 Unix 操作系统构造一个功能完善、操作简单以及界面友好的桌面环境，它是 GNU 计划的正式桌面。

3. Xfce

Xfce(XForms Common Environment)创建于 2007 年 7 月，类似于商业图形环境 CDE，是一个运行在各类 Unix 下的轻量级桌面环境。原作者 Olivier Fourdan 最先设计 Xfce 是基于 XForms 三维图形库。Xfce 设计的目的是用来提高系统的效率，在节省系统资源的同时，能够快速加载和执行应用程序。

1.4　Linux 的发行版本

不同的系统厂商在开发 Linux 时，虽然可能使用同一个 Linux 内核，但为了确立自己的品牌，都会使用不同的名称为这些发布的版本命名，这就是 Linux 的发行版本。其中，最著名的有 Red Hat 公司的 Red Hat 系列以及社区组织的 Debian 系列，还有 FC 系列、Ubuntu 系列等。下面，我们就来简单介绍一下目前比较流行的 Linux 发行版本。

1.4.1　Red Hat

Red Hat Linux 系统是美国 Red Hat 公司的产品，是相当成功的一个 Linux 发行版本，

也是目前使用最多的 Linux 发行版本。Red Hat 因其易于安装而闻名，在很大程度上减轻了用户安装程序的负担，其中 Red Hat 提供的图形界面安装方式非常类似 Windows 系统的软件安装，这对于那些 Windows 用户而言，几乎可以像安装 Windows 系统一样轻松安装 Red Hat 发行套件。

Red Hat 作为 Linux 的发行版本，开放源代码是与其他操作系统(如 Windows 等)相比具有的先天优势，有利于全世界范围内的软件工程师和相关技术人员共同开发。对于 Red Hat 来说，开放源代码已经不只是一个软件模型，这正是 Red Hat 的商业模式。因为 Red Hat 坚信只有协作，企业才能创造出非凡质量和有价值的产品。

在 Red Hat 公司工作的 300 名工程师中，有 6 名来自全世界最顶尖的 10 名 Linux 核心开发者，7 名来自全球最出色的 10 名 Linux 开发工具工程师。全世界，也许只有 Red Hat 公司能够把 Linux 和开源技术以及企业级的培训、技术支持和咨询融合得如此美妙。Red Hat 的培训及认证被认为是 Linux 认证的标准。Certification 杂志的最新调查显示，RHCE(Red Hat 认证工程师)认证被公认为是总体质量最高的国际 IT 认证。

Red Hat 已经为全球 30 万台服务器提供 500 万套软件。作为全球企业最重要的 Linux 和开源技术提供商，Red Hat 还是目前全球最先自负盈亏的 Linux 企业，是纳斯达克上市公司，银行现金高达 29 亿美元；Red Hat 也是唯一获得全球顶尖 ISV(独立软件提供商)广泛支持的 Linux 厂商，是 Compaq、Dell、IBM、Intel 等一流 IT 企业的合作伙伴。

实际上，除了 Red Hat 系列以外，Red Hat 还在间接地支持着两个源于 Red Hat 的发行版本：Fedora Linux 和 CentOS。

Fedora Linux(第七版以前为 Fedora Core)由 Fedora Project 社区开发，红帽公司赞助，目标是创建一套新颖、多功能并且自由开放源代码的操作系统。Fedora 基于 Red Hat Linux，在 Red Hat Linux 终止发行后，红帽公司计划以 Fedora 来取代 Red Hat Linux 在个人应用的领域；而另外发行的 Red Hat Enterprise Linux(Red Hat 企业版 Linux，RHEL)则取代 Red Hat Linux 在商业应用的领域。Fedora 的功能对于用户而言，是一套功能完备、更新快速的免费操作系统，而对赞助者 Red Hat 公司而言，它是许多新技术的测试平台，被认为可用的技术最终会加入到 Red Hat Enterprise Linux 中。

CentOS(Community ENTerprise Operating System)是来自于 Red Hat Enterprise Linux 依照开放源代码规定释出的源代码所编译而成。由于出自同样的源代码，因此要求对稳定性要求较高的服务器以 CentOS 替代商业版的 Red Hat Enterprise Linux 使用。两者的不同在于 CentOS 并不包含封闭源代码软件。

如图 1-2 所示是以上发行版本的产品标志。



图 1-2　Red Hat、Fedora、CentOS 的产品标志

1.4.2　Mandriva

Mandriva 是全球最优秀的 Linux 发行版本之一。2005 年之前稳居 Linux 排行榜第一

名。它是目前最易用的 Linux 发行版本，也是众多国际级 Linux 发行版本中唯一一个默认即支持中文环境的版本。它是法国 Mandriva 公司(前身为 Mandrake 公司)开发的 Linux 发行版本。Mandriva 公司现在仍然是欧洲最大的 Linux 厂商，Mandriva Linux 的前身为著名的 Mandrake Linux。Mandriva(Mandrake)项目是世界上第一个为非技术类用户设计的易于使用、安装和管理的 Linux 版本。Mandriva(Mandrake Linux)早期方便的字体安装工具和默认的中文支持，为 Linux 的普及作出了很大的贡献。现在的 Mandriva 系统是由 Mandrake 和 Conectiva 结合发展而来的。

Mandriva 的优点：友好的操作界面、图形配置工具，庞大的社区支持，NTFS 分区大小可变更。

Mandriva 的缺点：部分版本的问题较多，最新版本只发布给 Mandrake 俱乐部的会员。

如图 1-3 所示的是 Mandriva 的产品标志。



图 1-3　Mandriva 的产品标志

1.4.3　SUSE

SUSE 是德国最著名的 Linux 发行版本。它主要针对个人用户，在最初发行时，SUSE Linux 可以任意下载和安装，在 Novell 接手 SUSE Linux 的开发后，又发展创建了企业应用和高级桌面应用的 Linux 版本，包括以下版本：

- SUSE Linux Enterprise Server(SLES)
- Novell Open Enterprise Server
- Novell Linux Desktop

在 Novell 公司接手 SUSE Linux 的开发和发布进程后，它又资助了社区开发计划——openSUSE 项目。openSUSE 项目的主要目标是使 SUSE Linux 成为最易获得和最广泛使用的 Linux，成为最好的 Linux 用户桌面环境。

SUSE 的优点：专业、易用的 YaST 软件包管理系统。

SUSE 的缺点：FTP 发布通常要比零售版晚 1～3 个月。

SUSE 体验软件包管理系统：YaST (RPM)、第三方 APT (RPM)软件库(Software Repository)。

如图 1-4 是 SUSE Linux 和 openSUSE 的产品标志。





图 1-4　SUSE Linux 和 openSUSE 的产品标志

1.4.4 Debian

Debian GNU/Linux 是 1993 年由 Ian Murdock 发起的，受到当时 Linux 操作系统与 GNU 软件工程的影响，目标是成为一个公开的发行版。Debian 从一个小型紧密的自由软件黑客(Hacker)小组，逐渐成长为今日庞大且运作良好的开发者与用户社群。可以说，Debian 是现今为止最遵循软件工程项目规范的 Linux 操作系统。

Debian 系统分为三类版本：unstable、testing、stable。

- unstable 是最新的测试版本，包含很多最新的软件包。但是系统也不太稳定，适合桌面用户使用。
- testing 版本都经过 unstable 的测试，相对比较稳定，另外还支持不少新技术。
- stable 版本适用于服务器，包含的软件包大部分版本比较旧，但是稳定性和安全性都非常高。

Debian 的优点：遵循 GNU 规范，完全免费，优秀的社区资源。

Debian 的缺点：安装相对较难，stable 版本提供的软件相对过时。

如图 1-5 是 Debian 的产品标志。



图 1-5 Debian 的产品标志

1.4.5 Ubuntu

Ubuntu 是一个以桌面应用为主的 Linux 操作系统，是现在最流行的 Linux 桌面操作系统。其名称来自非洲南部祖鲁语或豪萨语的“ubuntu”一词(译为吾帮托或乌班图)，意思是“人性”、“我的存在是因为大家的存在”，是非洲传统的一种价值观，类似华人社会的“仁爱”思想。Ubuntu 基于 Debian 的 unstable 版本和 GNOME 桌面环境，与 Debian 的不同在于它每半年会发布一个新版本。Ubuntu 的目标在于为一般用户提供一个最新的，同时又相对稳定的主要由自由软件构建而成的操作系统。Ubuntu 具有庞大的社区力量，用户可以方便地从社区获得帮助。

Ubuntu 的安装非常人性化，与 Windows 一样简便易操作，只需要按照提示一步步进行安装即可，Ubuntu 被誉为是对硬件支持最好并且最完善的 Linux 发行版本之一，支持的软件也是最新的版本。

Ubuntu 的优点：为人气颇高的论坛提供优秀的资源和技术支持，具有固定的版本更新周期。

Ubuntu 的缺点：还未建立成熟的商业模式。

如图 1-6 是 Ubuntu 的产品标志。

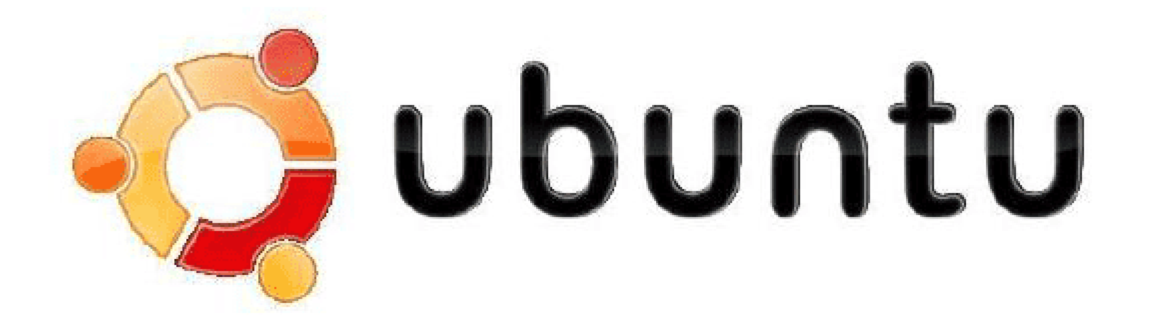


图 1-6　Ubuntu 的产品标志

1.4.6　Gentoo

Gentoo 是一个基于 Linux 的自由操作系统，它几乎能为任何应用程序或需求自动地做出优化和定制。追求极限的配置、性能，以及顶尖的用户和开发者社区，都是 Gentoo 体验的标志特点。Gentoo 的哲学思想是自由和选择，它得益于一种称为 Portage 的技术，Gentoo 能成为理想的安全服务器、开发工作站、专业桌面、游戏系统、嵌入式解决方案或者别的用户环境——你想让它成为什么，它就可以成为什么。

Gentoo 已经停止发布新的编译版，最近一次发布的版本是 10.0，之所以发行是为了纪念发行十周年。之所以不发行编译版，是因为没有必要，Gentoo 不提供传统意义的安装程序，他的安装光盘只提供一个 Linux 环境，从分区、挂载硬盘、下载编译内核，到书写 Grub 等都需要手动输入命令行一步步来操作。复杂的安装过程往往会让很多新手觉得沮丧，但是它确实能更好地帮你了解 Linux 的构建。

Gentoo 的优点：高度的可定制性、完整的使用手册、优秀的 Portage 系统。

Gentoo 的缺点：编译耗时多，安装速度较慢。

如图 1-7 是 Gentoo 的产品标志。



图 1-7　Gentoo 的产品标志

1.4.7　Slackware

Slackware Linux 是由 Patrick Volkerding 制作的 GNU/Linux 发行版，它是世界上使用最久的 Linux 发行版。在它的辉煌时期，曾经在所有发行版中拥有最多的用户数量。但是，伴随着 Linux 商业化的浪潮，Red Hat、Mandriva 和 SUSE 这些产品通过大规模的商业推广，占据了广大的市场；Debian 作为一个社区发行版，也拥有很大的用户群。相比之下，Slackware 的内敛，使得它从许多人尤其是使用 Linux 的新用户的视野中消失了。

但是，Slackware 依然是一套先进的 Linux 操作系统，为易用性和高稳定性双重目标而设计，它同时向新用户和高级用户提供一套完善的系统，可装备使用在从桌面工作站到服务器的任意系统环境，可以按需使用各种 Web、FTP 和 Email 服务器。

- Slackware 的优点：非常稳定、安全，始终坚持 Unix 的规范。
- Slackware 的缺点：所有的配置均需要通过编辑文件来进行，自动硬件检测能力较差。

如图 1-8 是 Slackware 的产品标志。



图 1-8 Slackware 的产品标志

1.4.8 红旗 Linux

红旗 Linux 是由北京中科红旗软件技术有限公司开发的一系列 Linux 发行版，包括桌面版、工作站版、数据中心服务器版、HA 集群版和红旗嵌入式 Linux 等产品，是中国较大、较成熟的 Linux 发行版之一。

- 红旗 Linux 的优点：中文支持能力优秀，适合亚洲人的使用习惯，优秀的服务器管理工具。
- 红旗 Linux 的缺点：对硬件的支持较差，桌面系统软件包安装不方便。

如图 1-9 是红旗 Linux 的产品标志。



图 1-9 红旗 Linux 的产品标志

1.5 本 章 小 结

本章首先介绍了 Linux 操作系统的产生环境和发展历程；然后，介绍了 Linux 内核版本及其桌面环境；最后，介绍了时下比较流行的 Linux 操作系统的各种发行版本及其优缺点。希望读者通过对本章的学习，能够了解 Linux 操作系统的特点及其适用的环境。

1.6 课 后 习 题

1. **填空题**

(1) Linux 操作系统是一种免费、______的______类操作系统，它继承了______功能强大、性能稳定等特点，具有良好的______特性，是现阶段服务器级操作系统的首选。

(2) GUI 的含义是_____。

2. 选择题

(1) 以下不属于 Linux 发行版本的是(　　)。

A. RedHat　B. 红旗 Linux　C. Unix

D. Windows　E. Ubuntu

(2) 以下属于 Unix 发行版本的是(　　)。

A. Slackware　B. Mandriva　C. Minix　D. Mac OS

(3) Linux 核心的许可证是(　　)。

A. NDA　B. GDP　C. GPL　D. GNU

3. 判断题

(1) Linux 是一种免费的完全的多任务操作系统，它完全运行在微处理器的保护模式下。Linux 完全兼容 POSIX.1 标准。

(2) 自由软件是指由开发者提供软件全部源代码并放弃包括版权在内的任何权利，任何用户都有权使用、复制、扩散、修改的软件，只要用户也将自己修改过的程序代码公开就行。

4. 简答题

(1) 简述 Linux 发行版本与内核版本的区别。

(2) 简述 Linux 的发展历史。

第 2 章

Linux 安装与桌面管理

本章开始介绍 Linux 服务器的安装，在安装过程中我们通过使用 VMware 虚拟机的方式安装操作系统，这种方式有很多的优点，便于练习和教学。

2.1 安装前的准备工作

在安装 Linux 服务器之前，我们首先要了解一些安装前的准备工作，包括 Linux 服务器的基本硬件要求，各种安装方式，以及如何制定分区策略等。

2.1.1 硬件要求

发布各种不同版本的 Linux 操作系统，根据其内核版本和所带的软件及其工具的不同，具体的硬件要求会稍有不同，但从总体上来说，基本硬件要求如下。

- CPU：Intel 386 及以上的处理器。
- 内存：至少 64MB，推荐使用 128MB 或者更多。
- 硬盘：系统分区需要分配 5～8GB 的空闲空间，例如 Red Hat Enterprise Linux 5 (RHEL5)完全安装需要 6～7GB 空间。
- 显卡：VGA 显卡。
- 光驱：CD-ROM 或者 DVD-ROM(根据安装介质的不同)。
- 其他设备：声卡、网卡等。

如果使用虚拟机的方式安装 Linux，对 CPU 以及内存的要求会更高一些。

2.1.2 安装方法

Linux 有许多安装方法，可以通过各种方式检索安装文件。例如，如果我们只在一台服务器中安装 Linux 操作系统，那么使用光盘安装是比较合适的方法，因为它需要的安装时间最短。然而，如果我们需要在一个网络中的上百台服务器中安装 Linux 操作系统，那么我们可以将必需的文件组织起来建立一个集中式安装源，这样就不需要在每台服务器中反复地插入、取出光盘，从而大大节约管理员的时间。要在所有的系统上同步安装，可以用 PXE 启动所有系统，而不用为每个系统单独制作一张光盘，它们可以使用网络共享一组安装文件来进行安装。

Linux 常用的安装方法有 5 种：从光盘安装、从硬盘安装、网络安装、kickstart 安装、PXE 安装。

1. 从光盘安装

用光盘安装是最直接的方法，在服务器的光驱中放入安装光盘，确认 BIOS 中设置为光盘启动，引导系统，然后管理员通过键盘操作，逐步设置安装过程中的选项。

2. 从硬盘安装

从硬盘安装需要首先在硬盘驱动器的分区中安装光盘的 ISO 镜像，从而使安装程序可以访问(格式必须为 ext2、ext3 或者 vfat)。在 RHEL5 中，还要求通过第一张安装光盘的 boot.iso 镜像文件来创建引导光盘。

3. 网络安装

通过网络安装非常适合一次在多台服务器中安装 Linux 操作系统的情况。从网络安装也需要通过 boot.iso 来创建引导光盘，或者从 PXE 引导。引导启动后，需要选择合适的网络安装方法(NFS、FTP 或者 HTTP)。系统必须能够通过选定的网络协议使用安装源。

4. kickstart 安装

kickstart 是 Red Hat 脚本化安装方法的名称。编写 kickstart 格式的脚本后，安装程序可以通过引导光盘或者 PXE 启动，然后选择 kickstart 脚本文件的位置。

5. PXE 安装

PXE 即 Pre-Execution Environment，现在绝大多数网卡都支持这种启动方式。它可以通过链接到网络文件服务器进行安装，并从网络中检索安装文件引导。

2.1.3　Linux 分区

在安装 Linux 之前，我们还应该了解 Linux 硬盘分区的相关知识。

现在流行的操作系统无一例外地使用了虚拟内存技术。Windows 系统使用交换文件来实现虚拟内存，而 Linux 使用了交换分区来实现。因此，在安装 Windows 时系统只需要使用一个分区就可以满足要求，但 Linux 系统至少需要两个分区：一个是系统分区，另一个是交换分区(也称作 swap 分区)。

提示： 当然，如果安装 Linux 系统的服务器具有海量的内存，可以不设置交换分区。但通常情况下，用户的服务器是无法满足大量交换数据所需的内存容量的(根据服务器应用的不同，物理内存需要达到几千兆甚至上百千兆才能满足要求)，所以绝大多数情况下需要设置交换分区。

1. 硬盘分区的基本概念

硬盘使用前需要进行分区。磁盘分区分为主分区、扩展分区和逻辑分区。一个硬盘最多只能有 4 个主分区。除了主分区，一个硬盘还可以划分一个扩展分区，一个硬盘只能有一个扩展分区，在这个扩展分区中又可以划分多个逻辑分区。

2. 硬盘分区的命名

在 Linux 操作系统中，所有的设备都是以一个文件的形式表示，硬盘也不例外。Linux 下的设备名存放在“/dev”目录中。

Linux 硬盘设备的命名规则如下：

服务器的第一块 IDE 接口硬盘对应文件为/dev/hda。

服务器的第二块 IDE 接口硬盘对应文件为/dev/hdb，依此类推。

服务器的第一块 SCSI 接口硬盘对应文件为/dev/sda。

服务器的第二块 SCSI 接口硬盘对应文件为/dev/sdb，依此类推。

为分区的命名使用“对应硬盘+分区数字”的形式来表示，例如：

服务器的第一块 IDE 接口硬盘的第 1 个分区对应文件为/dev/hda1。

服务器的第一块 IDE 接口硬盘的第 2 个分区对应文件为/dev/hda2，依此类推。

服务器第一块 SCSI 接口硬盘的第 1 个分区对应文件为/dev/sda1。

服务器第一块 SCSI 接口硬盘的第 2 个分区对应文件为/dev/sda2，依此类推。

在使用分区的数字编号时需要注意，数字 1～4 是留给主分区或者扩展分区使用的，逻辑分区的编号从 5 开始。

提示： IDE 接口和 SCSI 接口都是服务器常用的硬盘接口类型。IDE 在早期的服务器中经常使用，现在逐渐被更优秀的 SCSI 接口取代。在 Linux 中，不但 SCSI 接口的硬盘设备使用/dev/sd 开头，SAS/SATA/USB 接口的硬盘同样使用/dev/sd 开头。

3. Linux 分区与 Windows 分区的对比

在 Linux 中，我们是通过使用设备名(实质上为一个文件)来标识不同的硬盘和分区，这与 Windows 使用盘符来标识不同的硬盘和分区看似不同，但实质上，两者的基本思想是一致的，只是标识方法不同而已。如图 2-1 所示为 Linux 和 Windows 对硬盘进行分区的标识图。

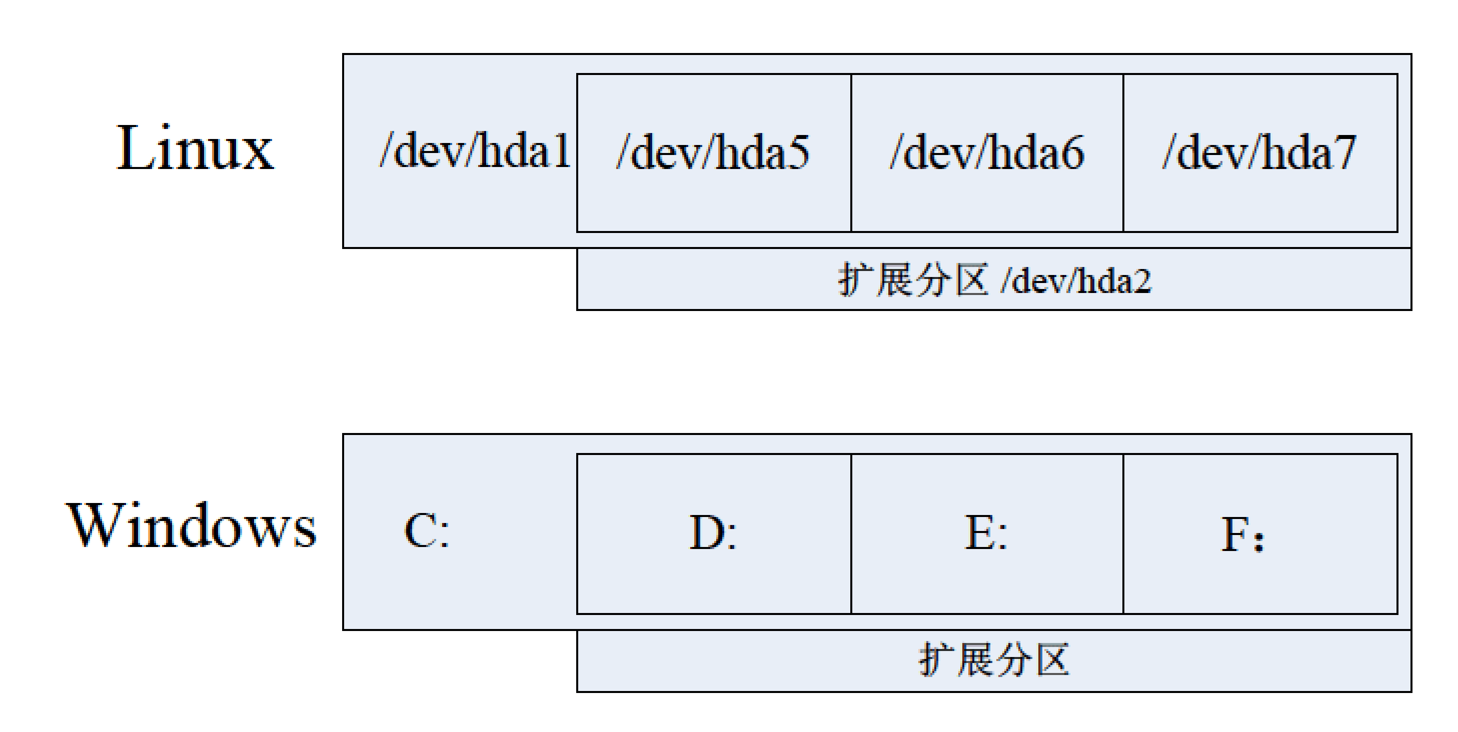


图 2-1　Linux 和 Windows 的分区标识方法对比

在 Linux 中，没有 Windows 中盘符的概念，所以要对硬盘中的某个分区进行操作的话，需要使用此分区的设备名。另外，在 Linux 操作系统上划分了分区以后，还需要在分区中创建文件系统，此操作相当于在 Windows 中格式化硬盘。区别在于 Windows 使用的文件系统类型为 FAT32 或者 NTFS 等；而 Linux 下经常使用的文件系统类型为 ext3、ReiserFS 等。

4. 静态分区和逻辑卷

在安装 Linux 之前，正确的分配各个分区的大小是一个非常困难的问题。系统管理员不但要考虑到当前系统所需的容量，还要预见该分区以后可能需要的容量。如果设计不合理，会导致某个分区容量不够用，系统管理员不得不调整分区的大小。

当使用静态分区时，如果要调整分区的大小，可以使用以下几种方法：

- 使用符号链接，这种方法将破坏 Linux 文件系统的标准结构。
- 使用调整分区大小的工具，这种方法需要停机后才能进行。

- 备份整个系统、重新对硬盘分区，然后再恢复数据到新的分区，这种方法更加麻烦，不但要停机终止服务器的服务，还需要重装系统。

从上面的几种方法我们可以看出，使用静态分区时的解决方法都不理想，所以在安装 Linux 系统时，我们要使用逻辑卷(Logical Volume)的分区方式。

逻辑卷(Logical Volume，LM)是一种建立在硬盘和分区之上的逻辑层，为文件系统屏蔽了下层分区的具体分布，从而提高了磁盘分区管理的灵活性。例如，将若干个磁盘分区连接为一整块的卷组(Volume Group)，形成一个存储池。管理员可以在卷组上随意创建逻辑卷组(Logical Volumes)，并进一步在逻辑卷组创建文件系统。管理员还可以对磁盘存储按照逻辑卷组的方式进行命名、管理和分配，而不需要再使用传统的“sda”和“sdb”等磁盘名。而如果服务器添加了新的磁盘，管理员也可以直接扩展文件系统，将新添加的磁盘纳入原有的逻辑卷中。

了解了逻辑卷的基本原理后，下面我们来看看逻辑卷的一些基本组成部分。

1) 物理卷(Physical Volume，PV)

物理卷是指整个硬盘、硬盘上的分区或者逻辑上与磁盘分区具有相同功能的设备。例如 RAID。它处于逻辑卷系统中的最底层，是逻辑卷的基本存储逻辑模块，和基本的物理存储介质不同的是，它包含与逻辑卷相关的管理参数信息。

2) 卷组(Volume Group，VG)

卷组建立在物理卷之上，由一个或者多个物理卷组成。可以将物理卷动态的添加到卷组中，管理员可以在卷组上创建一个或者多个逻辑卷。从功能上看，卷组相当于其他操作系统(例如 Windows)中的物理硬盘。

3) 逻辑卷(Logical Volume，LV)

逻辑卷处于卷组之上，创建的逻辑卷可以根据需要自由的扩展或者缩小其占用的磁盘空间。从功能上看，逻辑卷类似于其他操作系统中的磁盘分区，管理员可以在逻辑卷上创建文件系统(例如/home、/usr 等)。

4) 物理区域(Physical Extent，PE)

每个物理卷被划分为若干个基本单元，即物理区域，具有唯一编号的物理区域是可以被逻辑卷寻址的最小存储单元。物理区域根据实际情况在创建物理卷的时候设定，其默认值为 4MB，物理区域的大小一旦确定，就不能再改变，而且同一个卷组中的所有物理卷的物理区域大小必须一致。

5) 逻辑区域(Logical Extent，LE)

与物理卷类似，逻辑卷也被划为可寻址的若干个基本单元，称为逻辑区域，在同一个卷组中，物理区域和逻辑区域是相同的，一一对应。

图 2-2 描述了物理卷与逻辑卷之间的关系。

注意： 在 Linux 操作系统中，/boot 分区不能位于卷组中，因为引导装载的程序无法从逻辑卷中读取数据，所以在安装操作系统的时候，必须创建一个与卷组分离的/boot 分区。

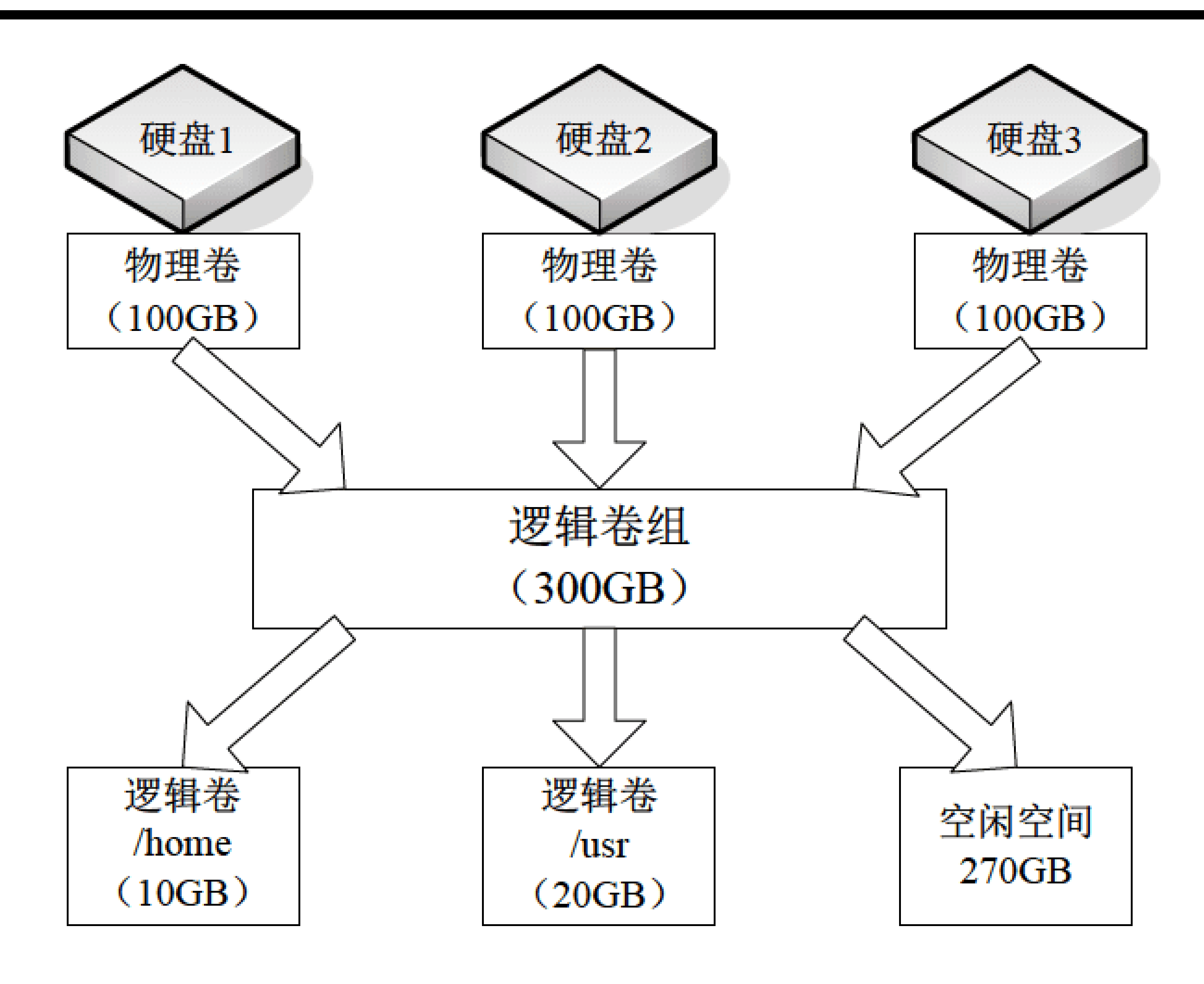


图 2-2　物理卷与逻辑卷的关系

2.2　VMware 虚拟机介绍

虚拟机(Virtual Machine)是指通过软件模拟的具有完整硬件系统功能的、运行在一个完全隔离环境中的完整计算机系统。通过虚拟机软件，用户可以在一台物理计算机上模拟出一台或多台虚拟的计算机，这些虚拟机完全就像真正的计算机那样进行工作，例如，可以安装操作系统、安装应用程序、访问网络资源等。对于真实主机而言，它只是运行在物理计算机上的一个应用程序，但是对于在虚拟机中运行的应用程序而言，它就是一台真正的计算机。因此，在虚拟机中进行软件评测时，可能系统一样会崩溃，但是，崩溃的只是虚拟机上的操作系统，而不是物理计算机上的操作系统，并且，使用虚拟机的“快照”功能，可以马上恢复虚拟机到安装软件之前的状态。

2.2.1　VMware Workstation 简介

VMware(威睿)是全球桌面到数据中心虚拟化解决方案的领导厂商。VMware Workstation 作为其主打的桌面虚拟机软件，提供用户可在单一的桌面上同时运行不同的操作系统，是进行开发、测试、部署新的应用程序的最佳解决方案。VMware Workstation 可在一部实体机器上模拟完整的网络环境，以及可便于携带的虚拟机器，其更好的灵活性与先进的技术胜过了市面上其他的虚拟计算机软件。对于企业的 IT 开发人员和系统管理员而言，VMware 在虚拟网路、实时快照、拖曳共享文件夹、支持 PXE 等方面的特点使它成为必不可少的工具。

VMware Workstation 允许操作系统和应用程序在一台虚拟机内部运行。虚拟机是独立运行主机操作系统的离散环境。在 VMware Workstation 中，用户可以在一个窗口中加载一台虚拟机，它可以运行自己的操作系统和应用程序。也可以在运行于桌面上的多台虚拟机之间进行切换，通过一个网络共享虚拟机(例如一个公司局域网)，挂起和恢复虚拟机以及退出虚拟机。

2.2.2　安装 VMware Workstation

在本书中，我们使用版本 VMware Workstation 7.1.3 来搭建测试环境，虽然 VMware Workstation 7.1.3 并不是最新版本，但其功能已经完全能够满足我们的要求，而且它对系统的要求更低一些。

安装 VMware Workstation 虚拟机软件的具体操作步骤如下。

(1) 双击 VMware Workstation 安装文件 VMware-Workstation-full-7.1.3-324285.exe，弹出 Workstation 7.1 安装界面，如图 2-3 所示。安装程序需要先自行解压安装包中的文件。

(2) 安装程序解压完成后，进入欢迎界面，如图 2-4 所示，单击 Next 按钮继续。

(3) 进入选择安装类型界面，如图 2-5 所示，选择 Typical(典型)安装。

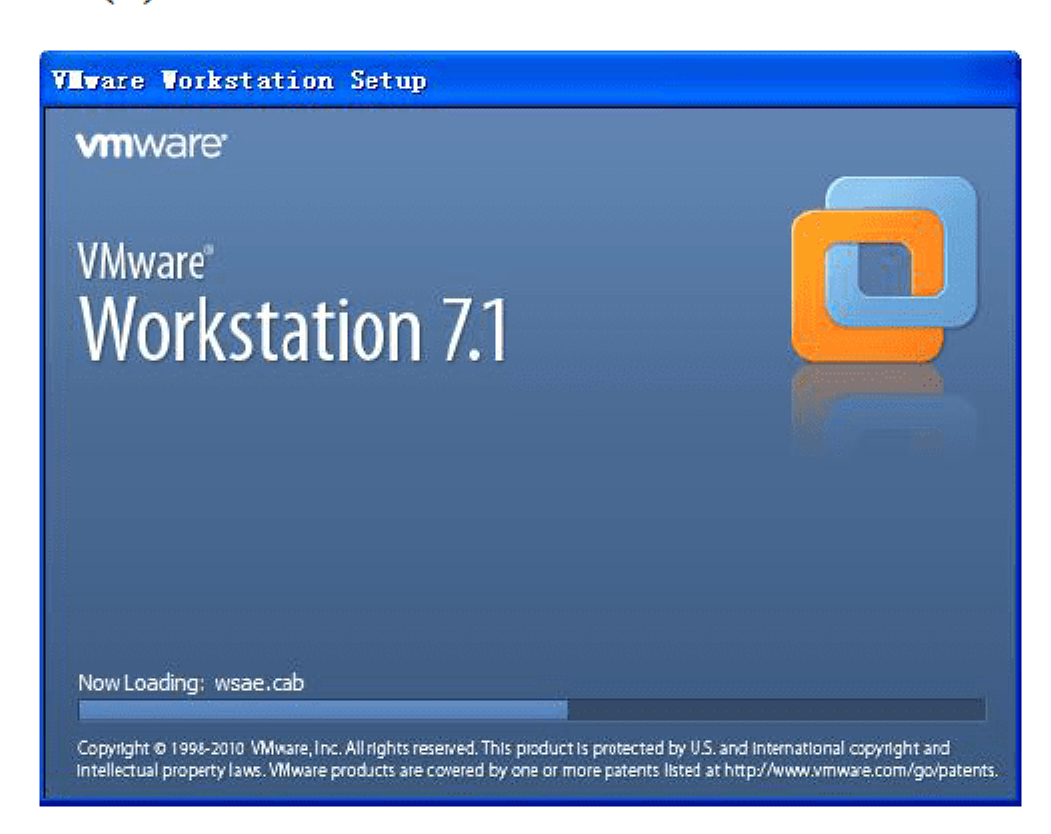


图 2-3　启动 VMware Workstation 安装程序

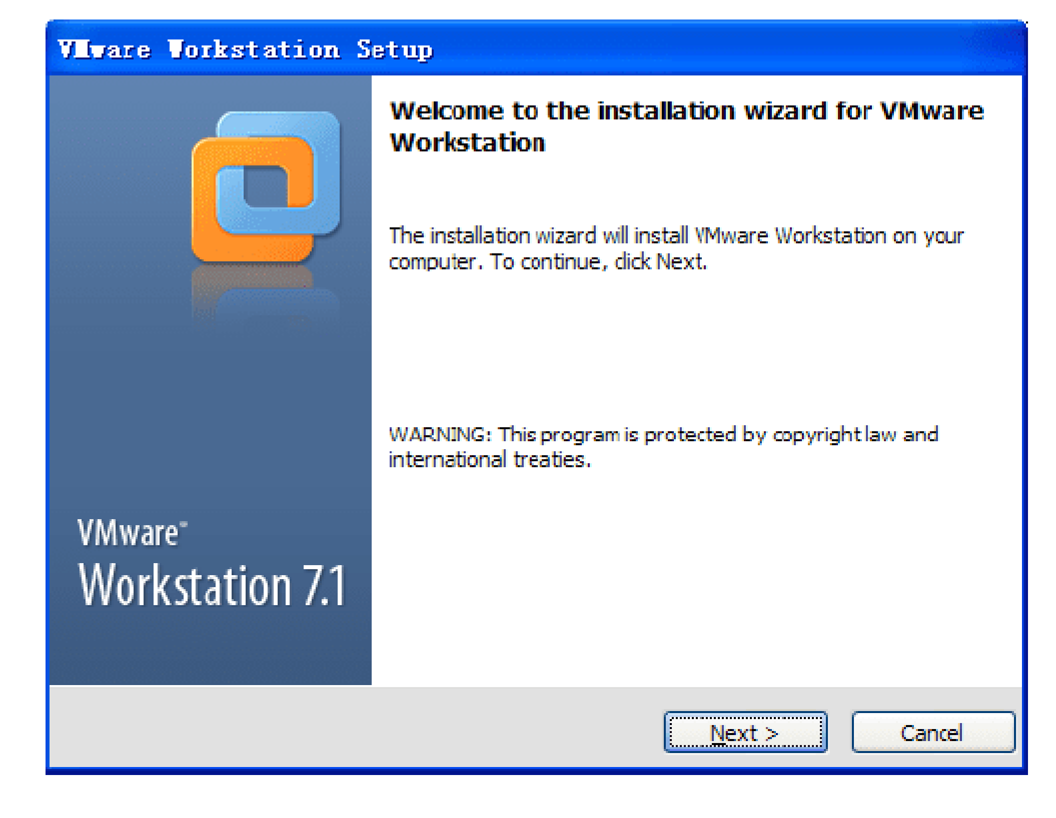


图 2-4　安装欢迎界面

(4) 在进入的界面中可以选择安装的路径，如图 2-6 所示。如果用户需要修改 VMware Workstation 程序的安装位置，可以单击 Change 按钮，在弹出的路径选择对话框中选择需要安装的位置；如果不需要改变安装位置，直接单击 Next 按钮。

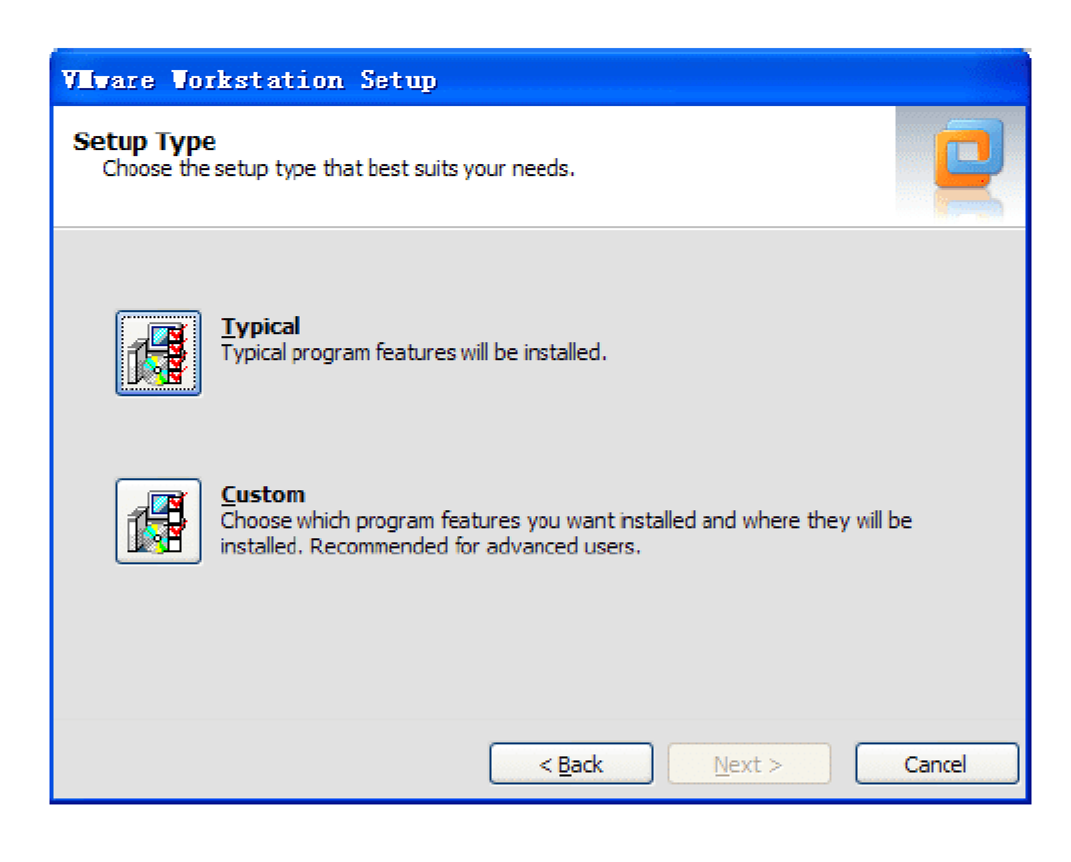


图 2-5　选择安装类型

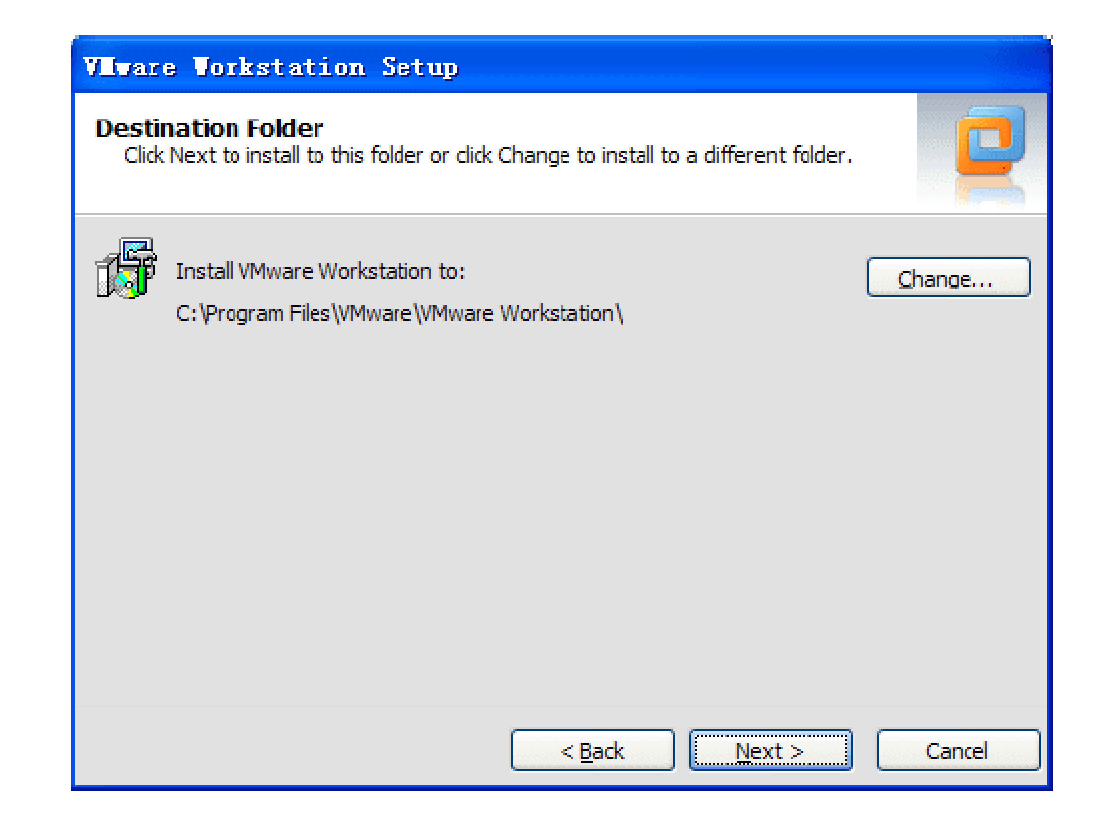


图 2-6　选择安装路径

(5) 进入软件升级界面如图 2-7 所示。如果用户希望时刻保持 VMware Workstation 程序的版本最新，可以选中 Check for product updates on startup 复选框，否则需要取消选中此选项。为了避免更新带来的升级风险，推荐取消此功能，当用户确实有需要更新虚拟机

版本的时候，可以采用手动更新的方法。单击 Next 按钮。

(6) 进入用户体验改进程序界面，如图 2-8 所示。如果用户允许此功能，VMware Workstation 将会上传用户使用的信息给 VMware 公司以改进用户体验，此处推荐开启此功能，单击 Next 按钮。

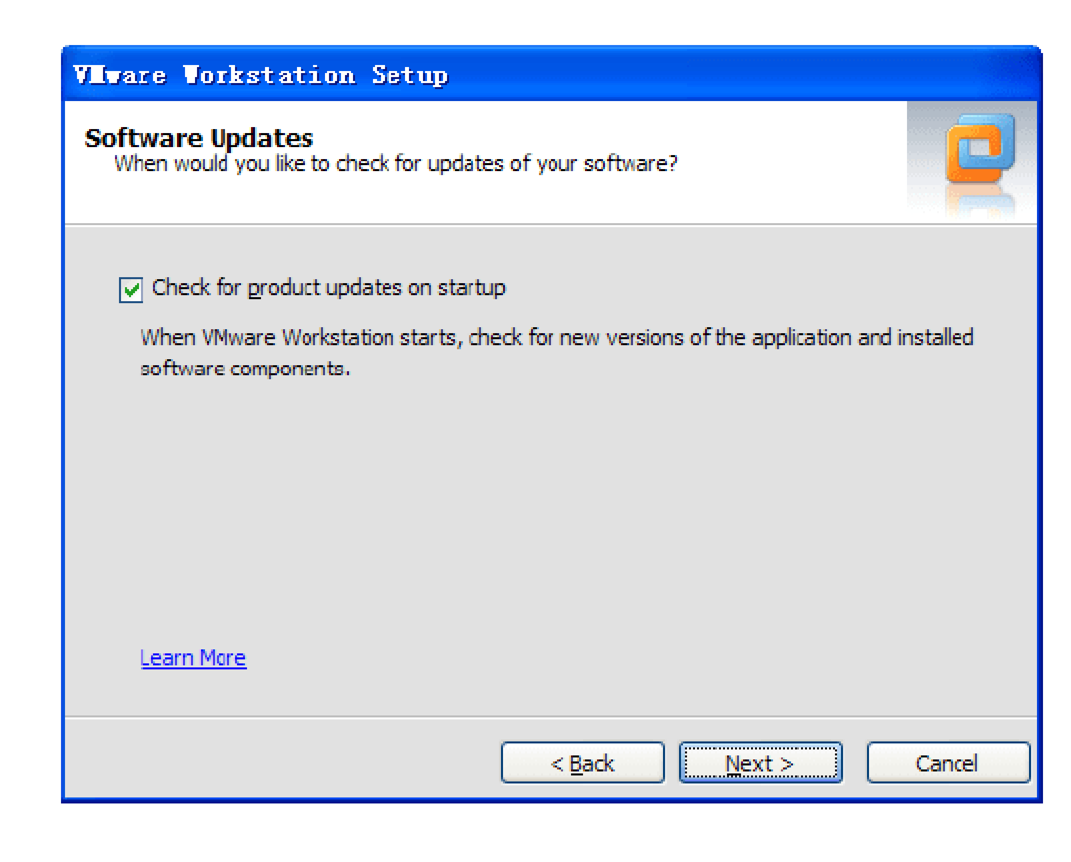


图 2-7 软件升级界面

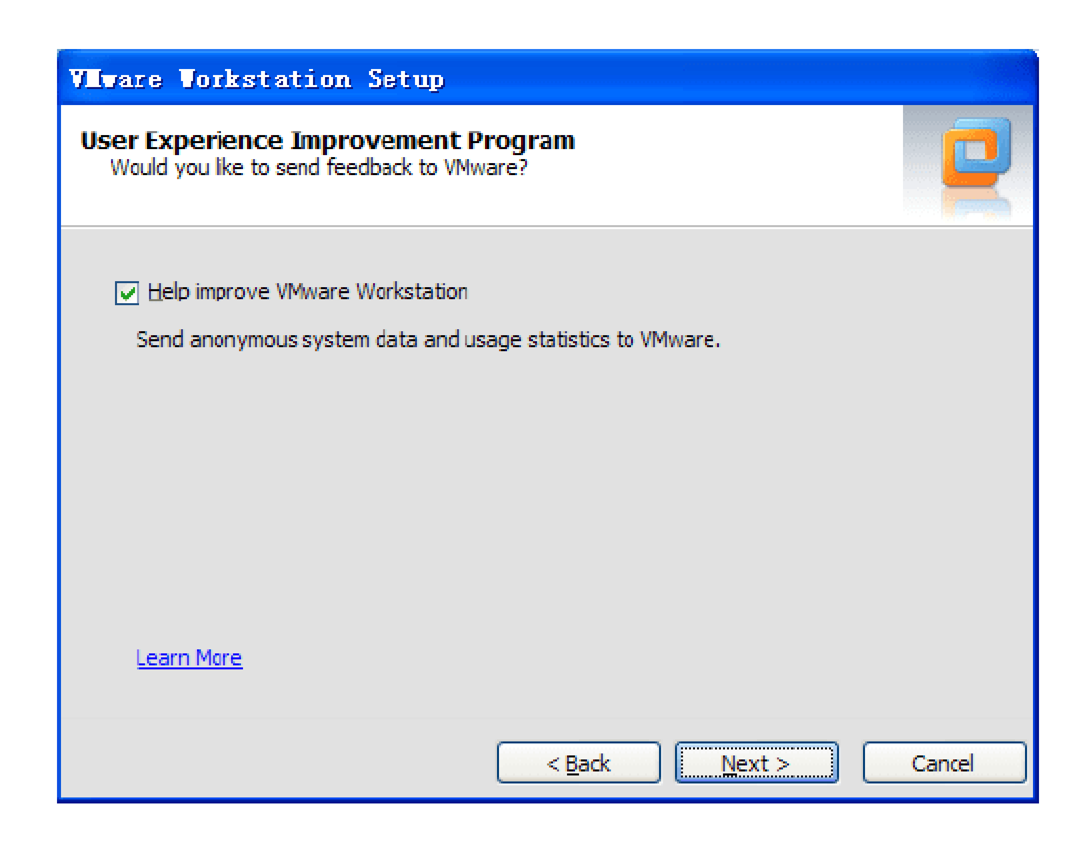


图 2-8 用户体验改进程序界面

(7) 进入快捷方式界面，如图 2-9 所示。此处用户可以选择安装程序是否在 Desktop (桌面)、Start Menu Programs folder(开始菜单)和 Quick Launch toolbar(快捷启动栏)中创建 VMware Workstation 的启动快捷方式，用户可以根据自己的需要和习惯设置，单击 Next 按钮。

(8) 安装程序将进行最后的确认，如图 2-10 所示。如果用户单击 Continue 按钮，安装程序将开始安装 VMware Workstation 到计算机，所有安装设置将不能再修改；如果希望修改安装设置，可以单击 Back 按钮来回顾修改刚才的设置。确认无误后，单击 Continue 按钮继续。

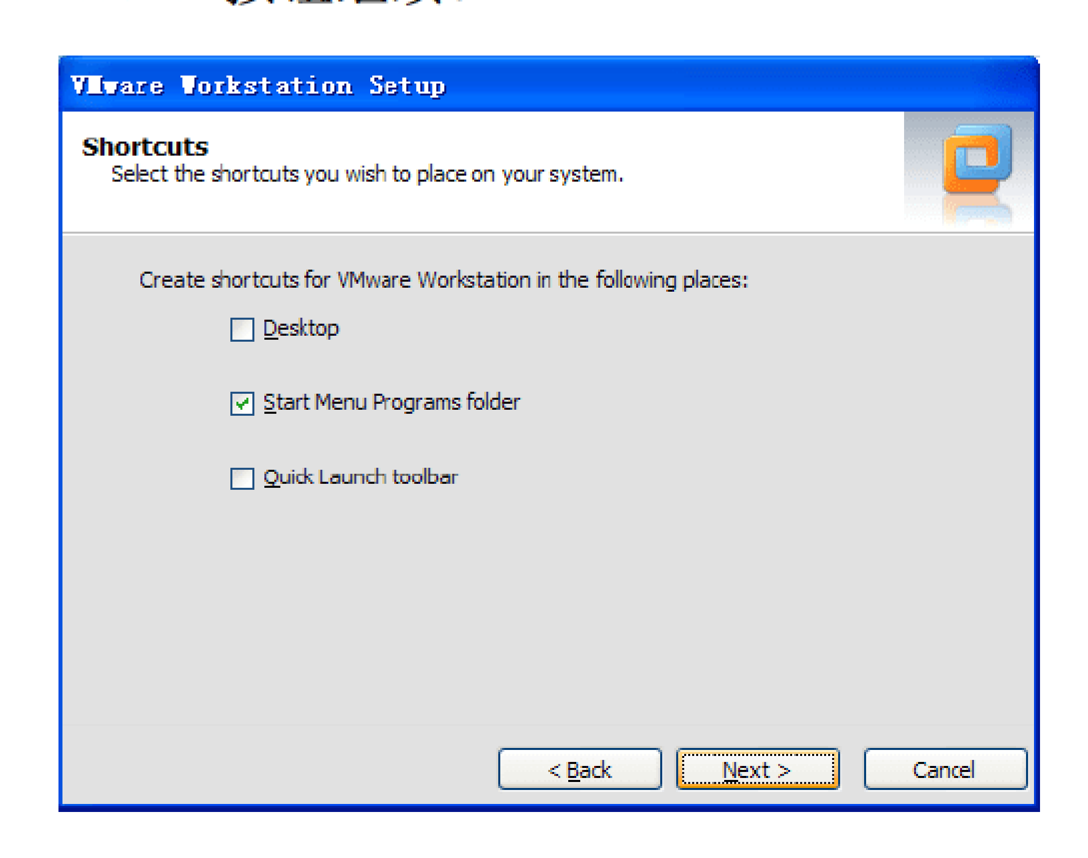


图 2-9 快捷方式界面

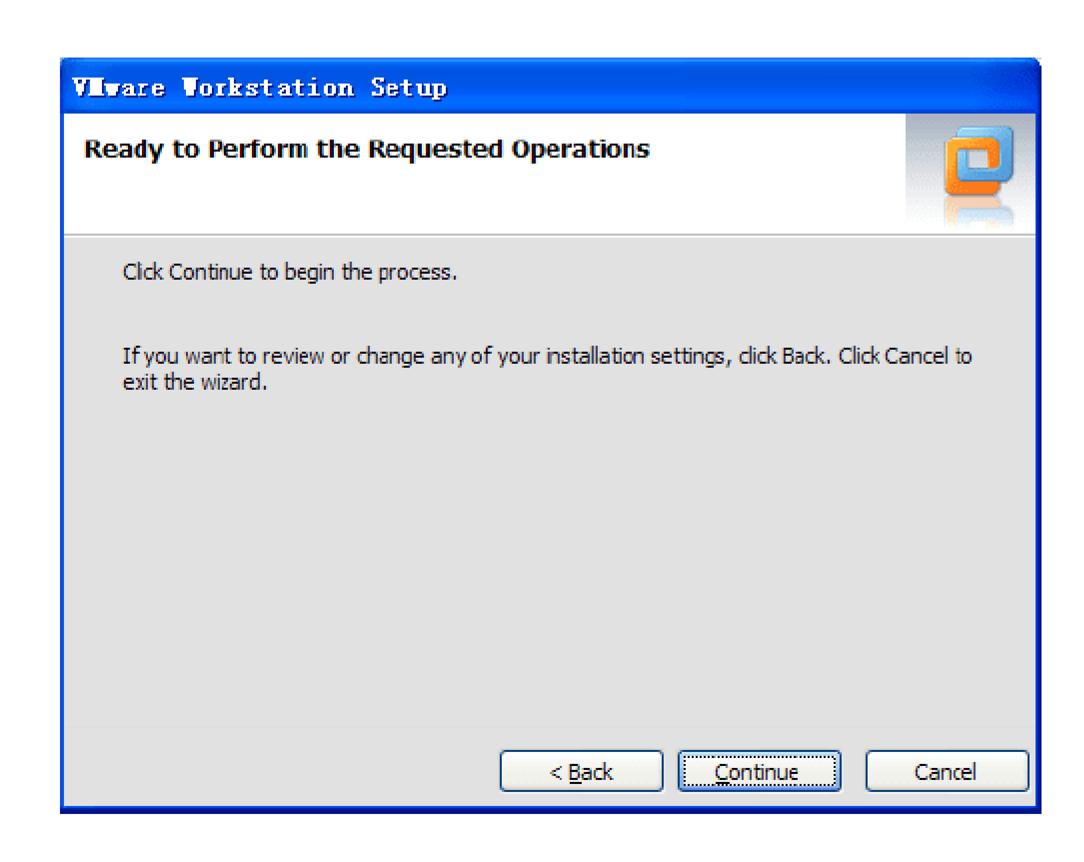


图 2-10 最后确认界面

(9) 安装程序开始安装 VMware Workstation，如图 2-11 所示。经过一段时间的等待，安装完成后程序会提示需要重新启动计算机，如图 2-12 所示。单击 Restart Now 按钮重启计算机，完成安装。

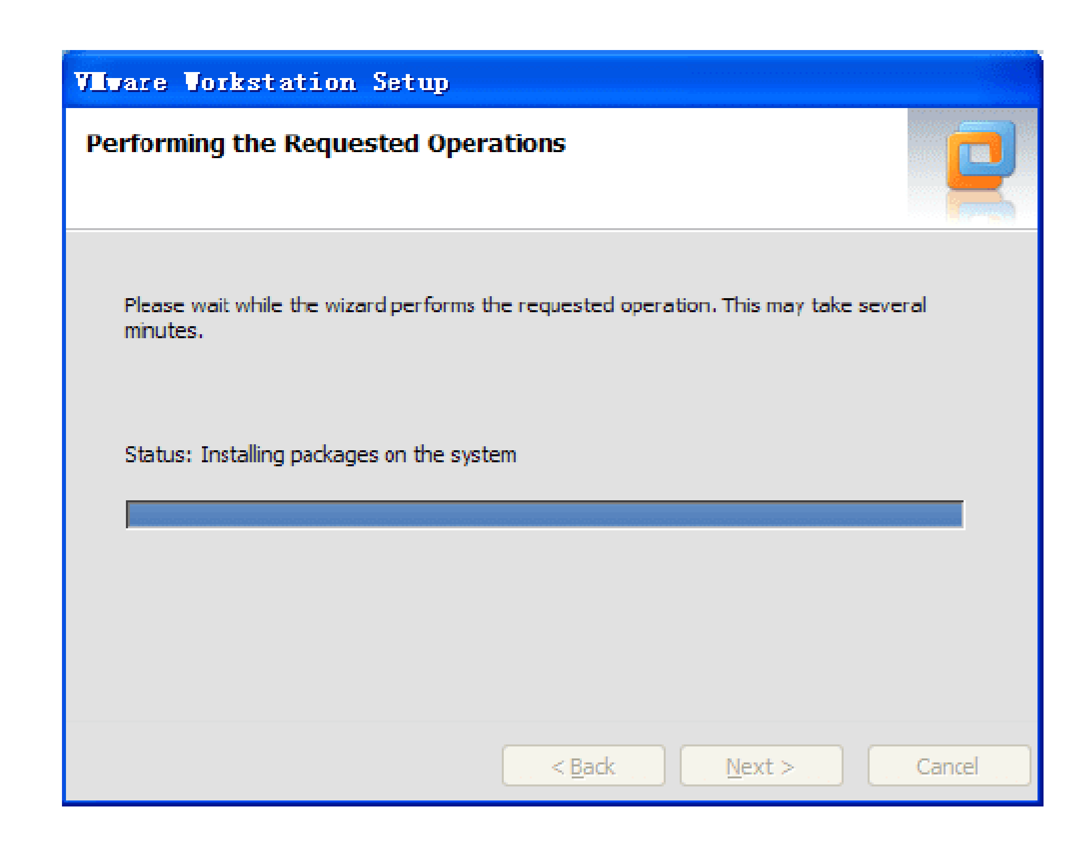


图 2-11　开始安装界面

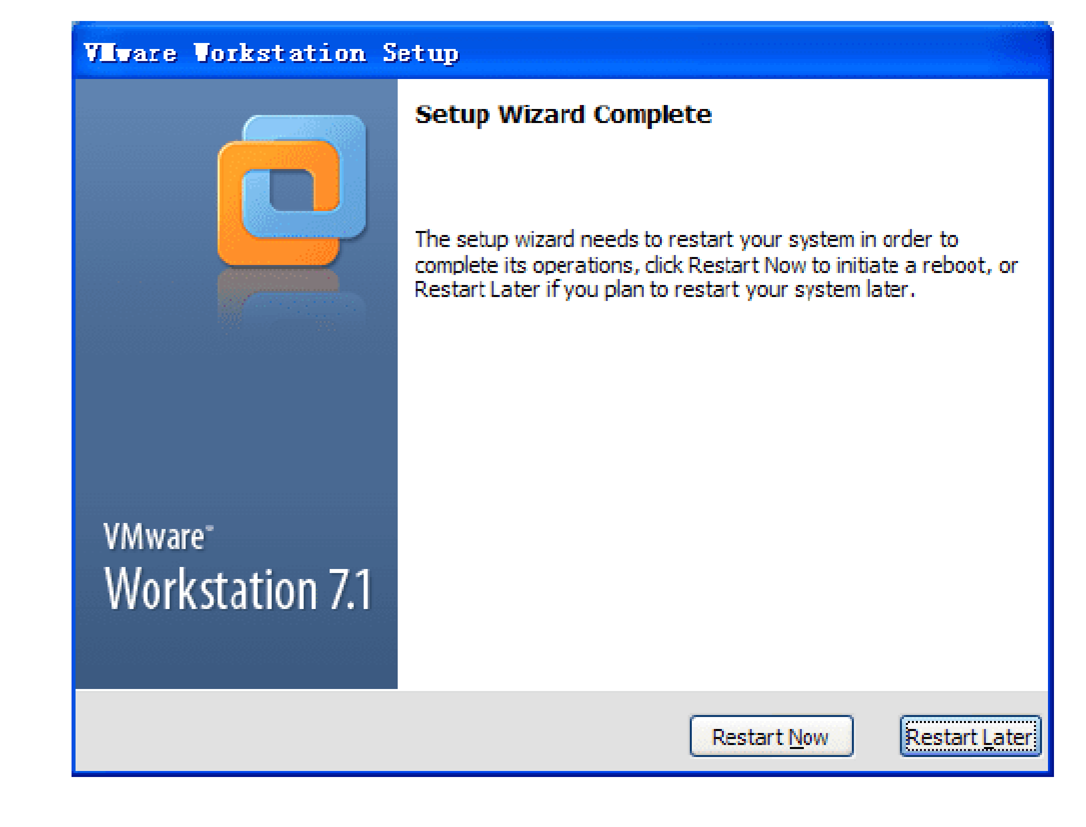


图 2-12　安装完成界面

2.3　安装 Linux 操作系统

虚拟机 VMware 安装完成后，我们就可以创建一个新的虚拟机并且在其中安装 Linux 操作系统了。

在本节中，我们使用 CentOS 5.5 版本为例讲解 Linux 操作系统的安装过程。在第 1 章中我们已经介绍过，CentOS 是来自于 Red Hat Enterprise Linux 依照开放源代码规定释出的源代码所编译而成。由于出自同样的源代码，因此除了一些由于版权问题而修改的商标以及显示界面外，CentOS 可以完全认为是商业版 Red Hat Enterprise Linux 的复刻版本。两者的最大不同在于 CentOS 并不包含封闭源代码软件。

安装过程中使用的光盘镜像文件名为 CentOS-5.5-i386-bin-DVD.iso。虽然我们是使用光盘镜像进行安装，但由于在虚拟机中载入光盘镜像相当于在虚拟光驱中放入光盘，所以下面我们介绍的安装方法属于光盘安装，而非硬盘安装。

2.3.1　创建新的虚拟机

在安装 CentOS 之前，先要使用 VMware 创建一个新的虚拟机，具体的操作步骤如下。

(1) 打开 VMware，单击 New Virtual Machine 按钮，如图 2-13 所示。

(2) 弹出新建虚拟机对话框，如图 2-14 所示。选择 Typical(典型)的安装方法，单击 Next 按钮。

(3) 进入客户操作系统安装界面，如图 2-15 所示。选择 Installer disc image file(iso)(从光盘镜像安装)，单击 Browse 按钮，选择 CentOS 的光盘镜像文件，确定后，我们会看到虚拟机自动显示“CentOS detected”提示，说明光盘镜像中的操作系统已经被虚拟机检测到，单击 Next 按钮。

(4) 进入简易安装信息界面，如图 2-16 所示。用户需要添加虚拟机的基本信息，包括服务器名、用户名及密码。添加完成后，单击 Next 按钮。

(5) 进入虚拟机名称界面，如图 2-17 所示。用户可以选择虚拟机文件保存的位置，单击 Next 按钮继续。

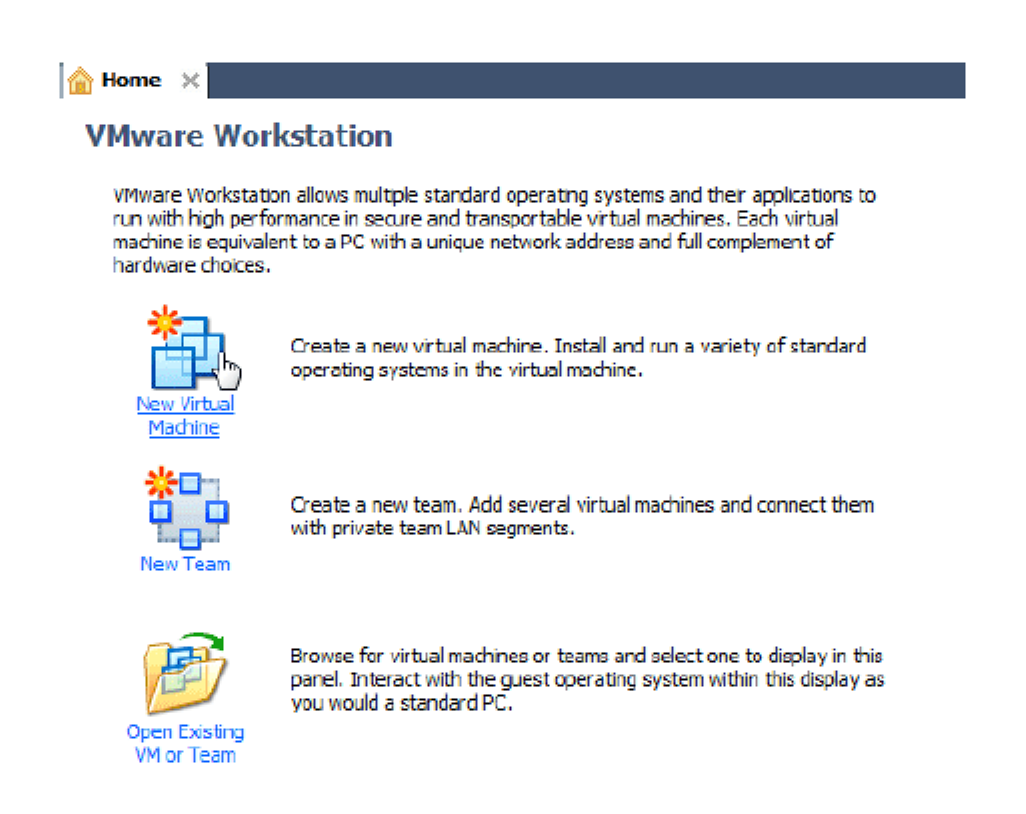


图 2-13 新建虚拟机



图 2-14 新建虚拟机对话框

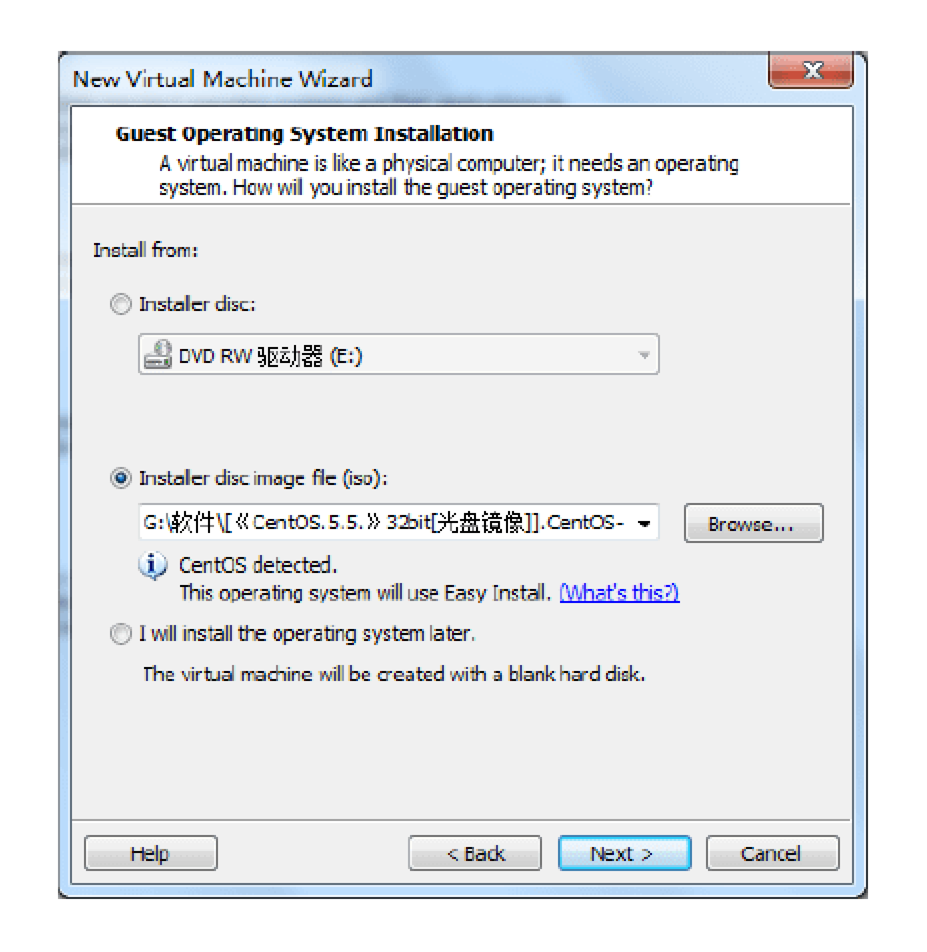


图 2-15 选择镜像路径

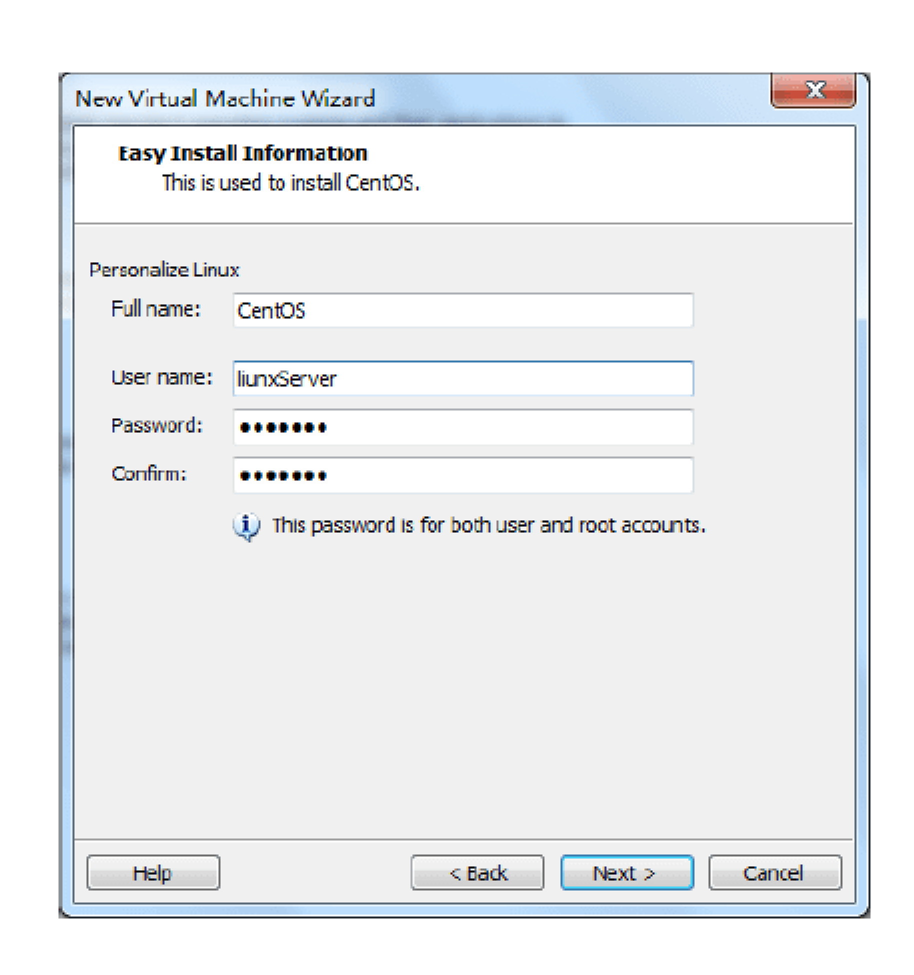


图 2-16 设置服务器及用户名

(6) 进入设置磁盘空间界面，如图 2-18 所示。用户可以设置虚拟机磁盘的大小，以及设置虚拟机磁盘文件时保存为单个文件还是分为若干个文件，一般情况下，为了便于管理和移植，都建议将磁盘设置保存为单个文件。单击 Next 按钮继续。

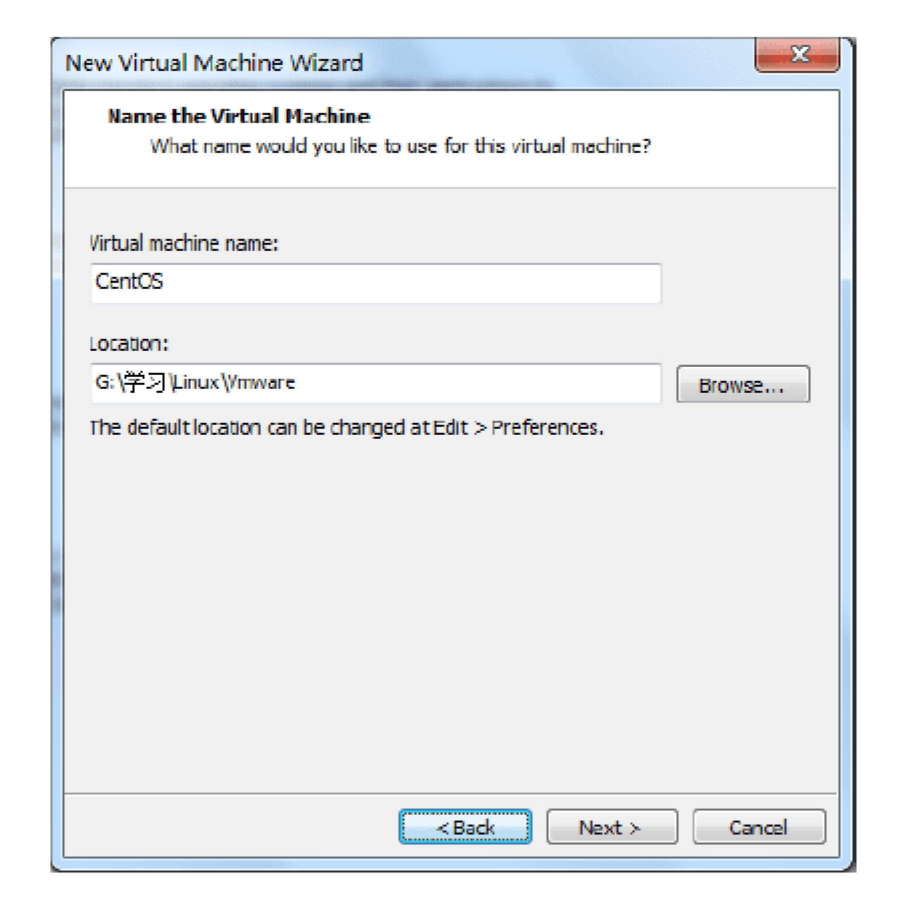


图 2-17 设置虚拟机文件保存路径

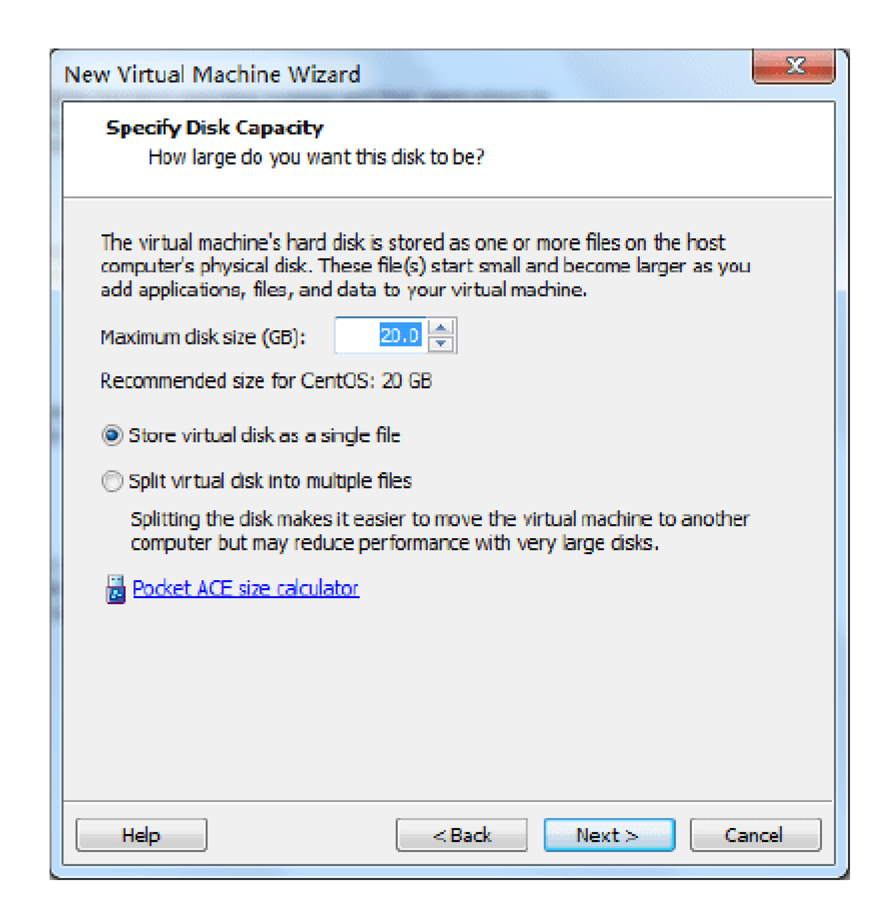
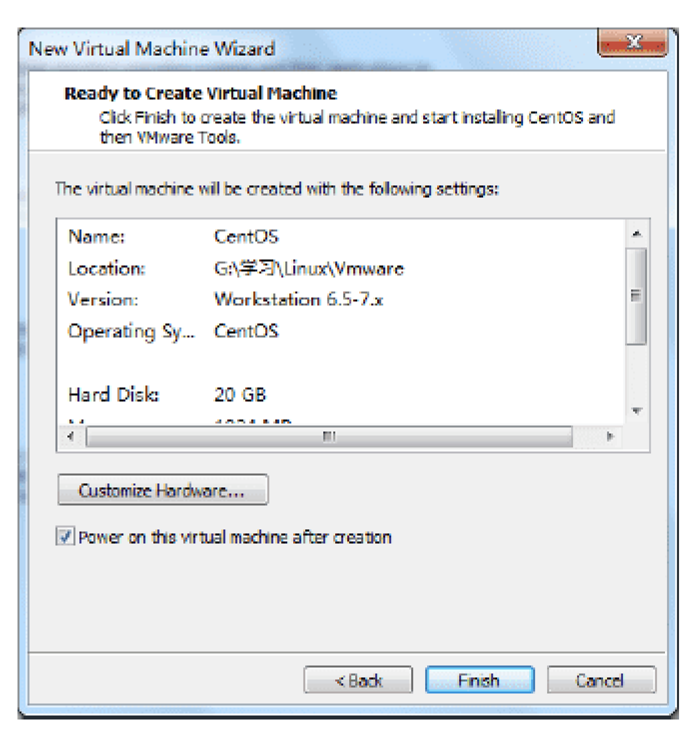


图 2-18 设置虚拟机磁盘大小

(7) 进入准备安装界面，如图 2-19 所示。用户可以在列表框中确认所有的设置信息，如果用户需要在创建完虚拟机后立即安装操作系统，可以选中 Power on this virtual machine after creation 复选框。最后，单击 Finish 按钮，虚拟机创建完毕。

(8) 创建完成后的虚拟机会显示在 VMware 软件的初始界面中，如图 2-20 所示。



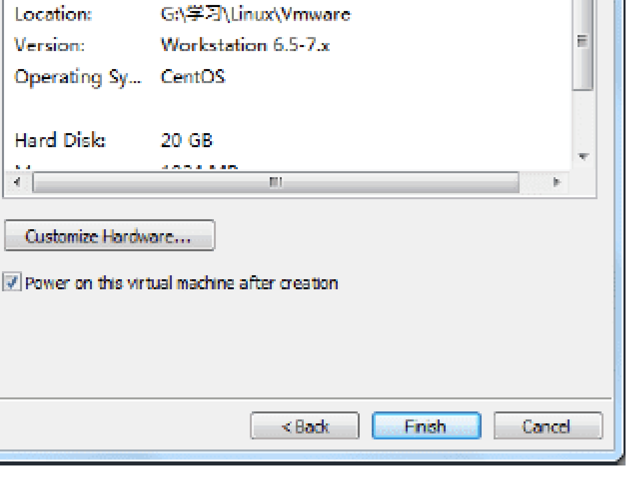


图 2-19　确认安装信息

图 2-20　新添加的虚拟机

2.3.2　开始安装 Linux

创建完虚拟机后，单击此虚拟机界面中的 Power on this virtual machine(打开虚拟机)命令，即可开始安装 Linux 操作系统，安装的具体操作步骤如下。

(1) 经过短暂的系统基本信息载入，我们可以看到 CentOS 安装起始界面如图 2-21 所示。一共有 3 种安装方式：

- 在图形界面中安装，直接按 Enter 键即可。
- 在文字界面中安装，需要输入 Linux text，然后按 Enter 键。
- 查看操作系统的其他信息按对应的功能键即可，包括显示主界面(F1 键)、显示选项(F2 键)、帮助信息概述(F3 键)、显示 Linux 核心信息(F4 键)、修复操作系统(F5 键)。

为了让读者能够更加深入了解 Linux 操作系统的安装过程，我们使用文本界面的安装方式，输入 Linux test，再按 Enter 键。

(2) 经过短暂的等待，操作系统载入安装信息，进入如图 2-22 所示的界面。提示用户系统找到的光盘介质，选择 OK 按钮通过光盘安装操作系统，开始检测光盘内容，如图 2-23 所示。

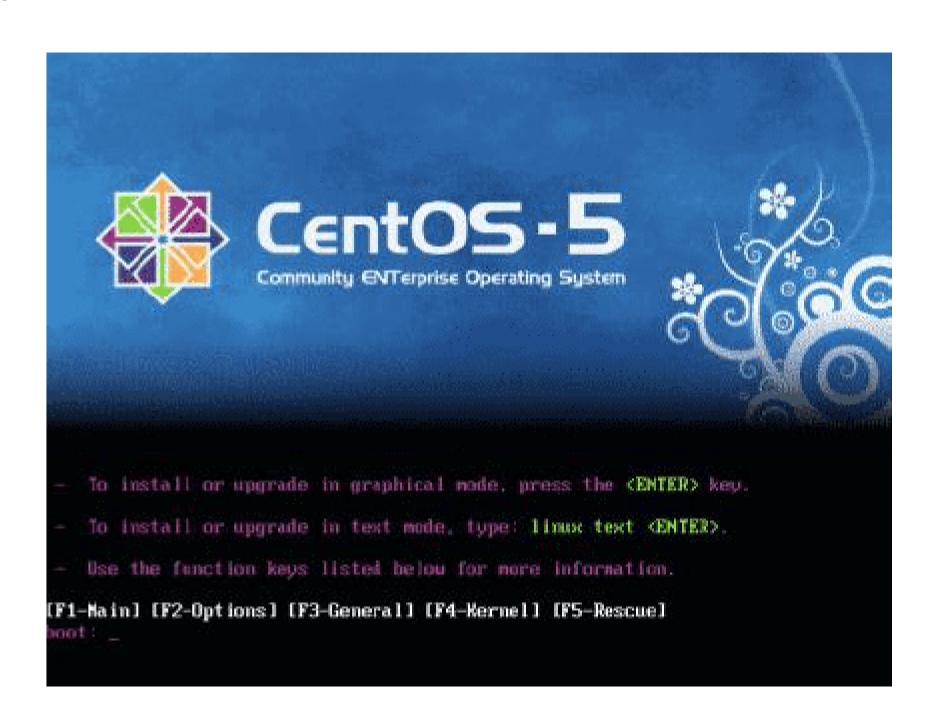


图 2-21　安装起始界面



图 2-22　检测光盘

(3) 检测完成后，进入语言选择界面，如图 2-24 所示。由于文本安装模式不支持中文，我们选择默认的英文即可，选择 OK 按钮继续。

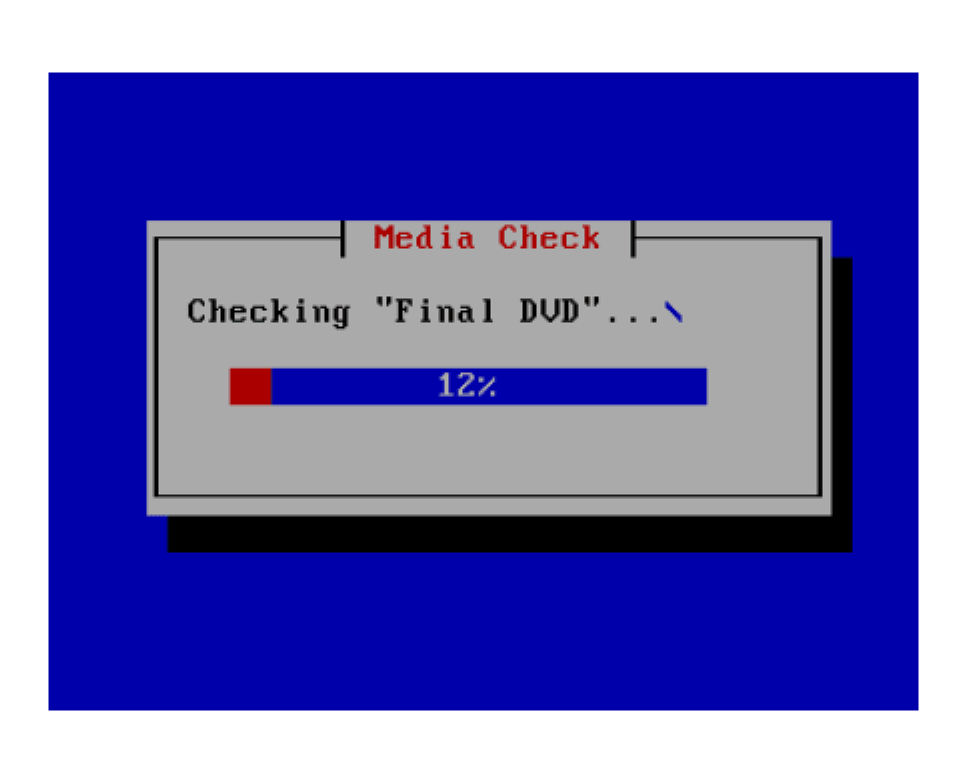



图 2-23　检测光盘



图 2-24　选择语言

(4) 进入键盘选择界面，如图 2-25 所示。使用默认设置即可，选择 OK 按钮继续。

(5) 进入分区类型界面，如图 2-26 所示。一共有 4 种分区方式可以选择：

- 删除所有的现有分区，创建默认的分区布局。
- 删除所有的 Linux 分区，创建默认的分区布局。
- 使用空闲空间创建默认分区布局。
- 自定义分区布局。

由于是全新的系统安装，我们选择第一种方式，选择 OK 按钮继续。

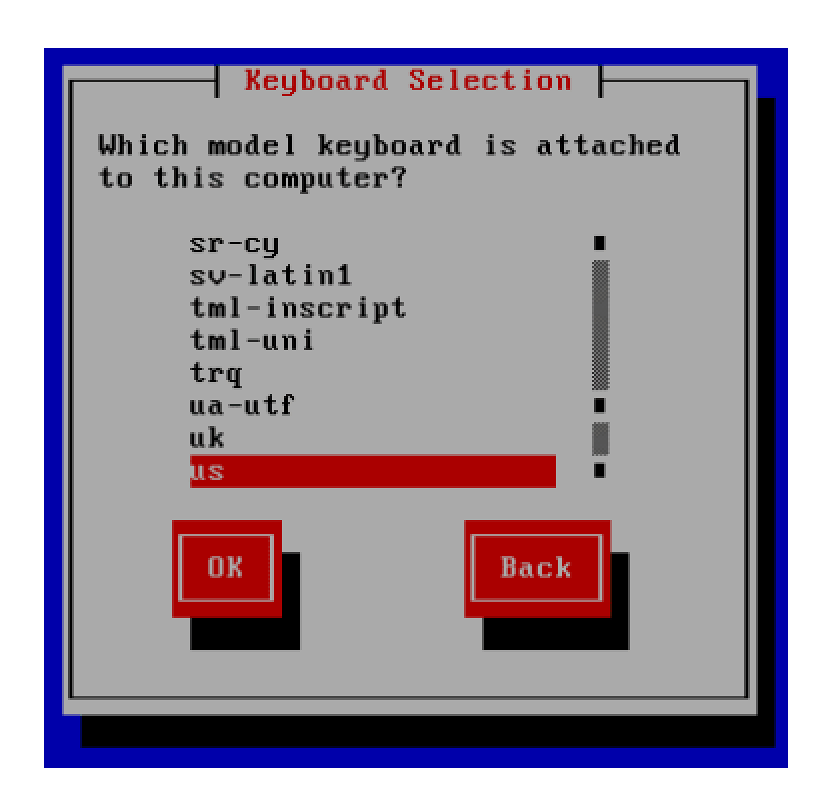


图 2-25　键盘选择

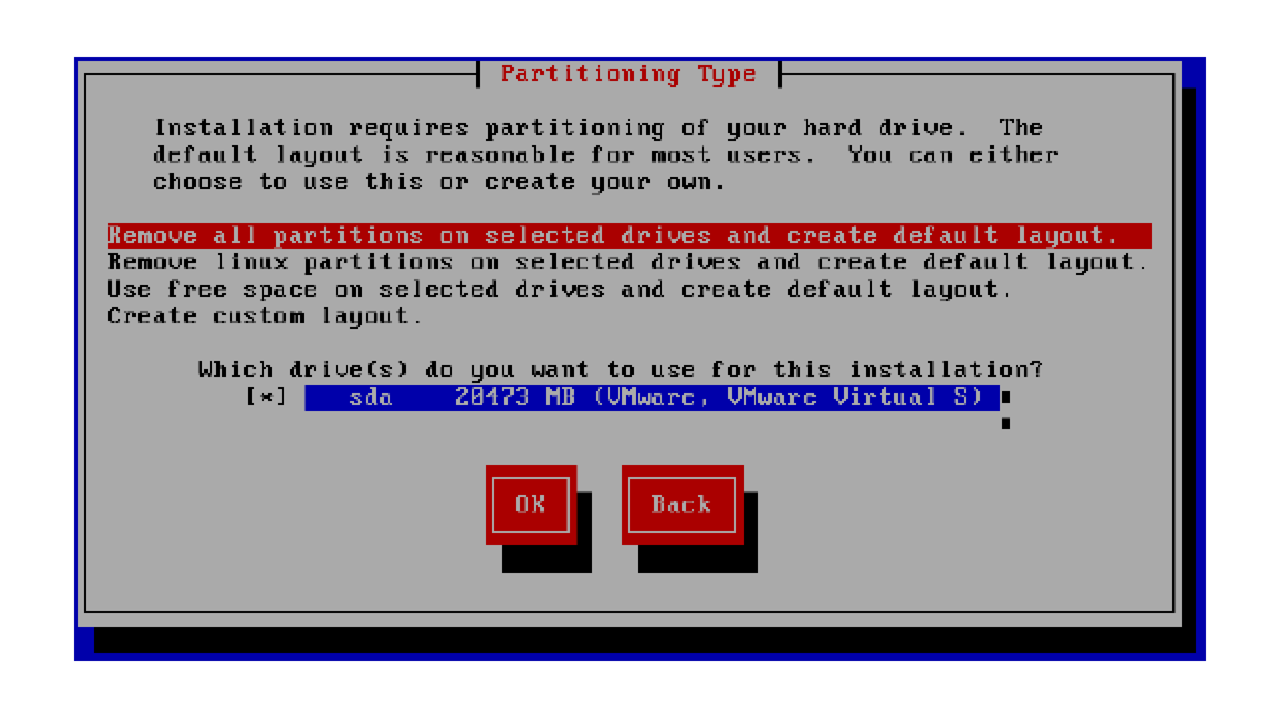


图 2-26　选择分区方式

(6) 系统会弹出警告界面提示用户将删除所有的分区，选择 OK 按钮即可。然后，进入了分区界面，可以查看默认的分区布局，如图 2-27 所示。此默认分区布局的特点是：

- 单独创建了/boot 分区，分区类型为 ext3。
- 将所有磁盘容量创建为一个物理卷。
- 将此物理卷加入到一个名为 VolGroup00 的卷组中。
- 在此卷组中分别创建了两个逻辑卷：LogVol00 和 LogVol01，分区类型分别为 ext3 和 swap。

从 2.1.3 节中的介绍可知，这种默认分区的方式是不安全的，所有的数据都被存放在

唯一的一个基于卷组的文件系统中，一旦这个卷组出现问题，其中的所有数据将不能被读取，很可能造成系统不能正常启动。安全的分区方案应该是将系统数据与普通用户的数据分离，也就是创建一个单独的/home 逻辑卷，并且应该将不经常变化的系统数据与经常变化的系统数据分离，即分别创建/usr 和/var 逻辑卷。

根据以上原则，我们将默认分区布局修改为如图 2-28 所示的样子。修改完成后，选择 OK 按钮继续。

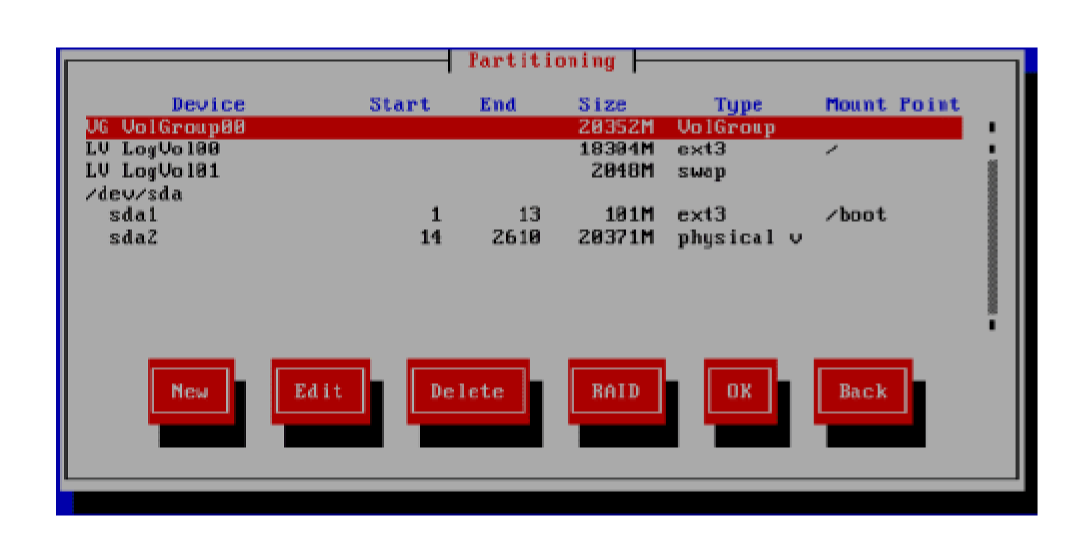


图 2-27　查看默认分区

图 2-28　修改默认分区

(7) 进入启动加载器设置界面，如图 2-29 所示。使用默认的 GRUB 启动加载器，选择 OK 按钮继续。

(8) 系统出现提示信息，如图 2-30 所示。告知用户有些系统在安装过程中需要设置特殊参数，在本安装中不需要用到，直接选择 OK 按钮继续。

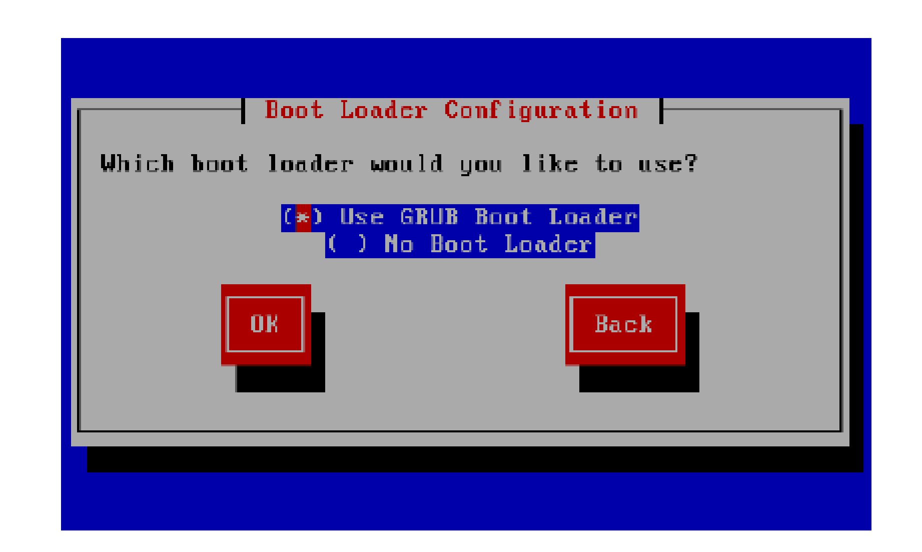


图 2-29　选择启动加载器

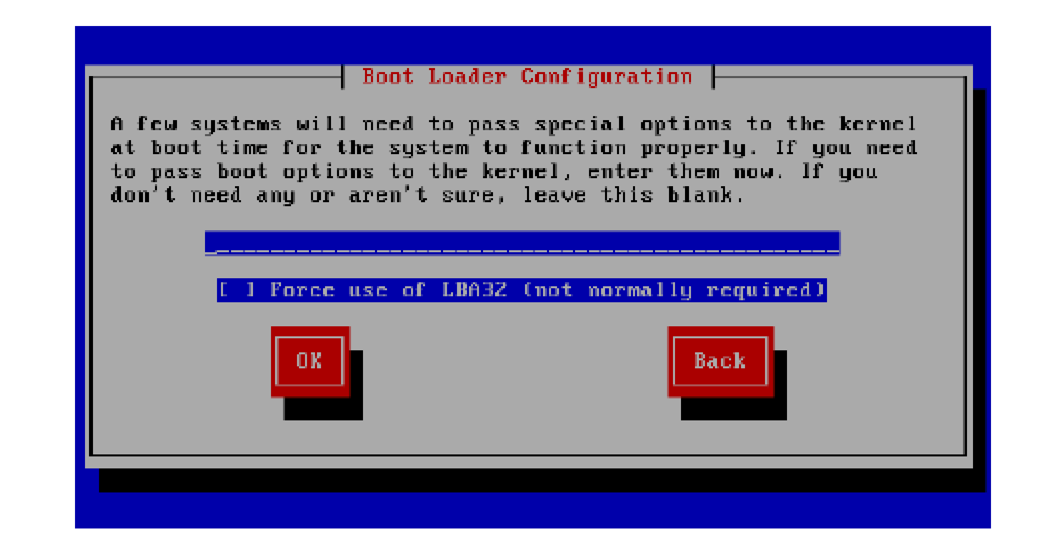


图 2-30　输入系统启动参数

(9) 进入设置 GRUB 密码界面，如图 2-31 所示，此处为了简化步骤，不设置 GRUB 密码，直接选择 OK 按钮继续。

(10) 进入设置启动卷界面，如图 2-32 所示。如果服务器安装有多个 Linux 操作系统，需要选择 Linux 的默认启动卷，在本例中，使用默认值即可，单击 OK 按钮继续。

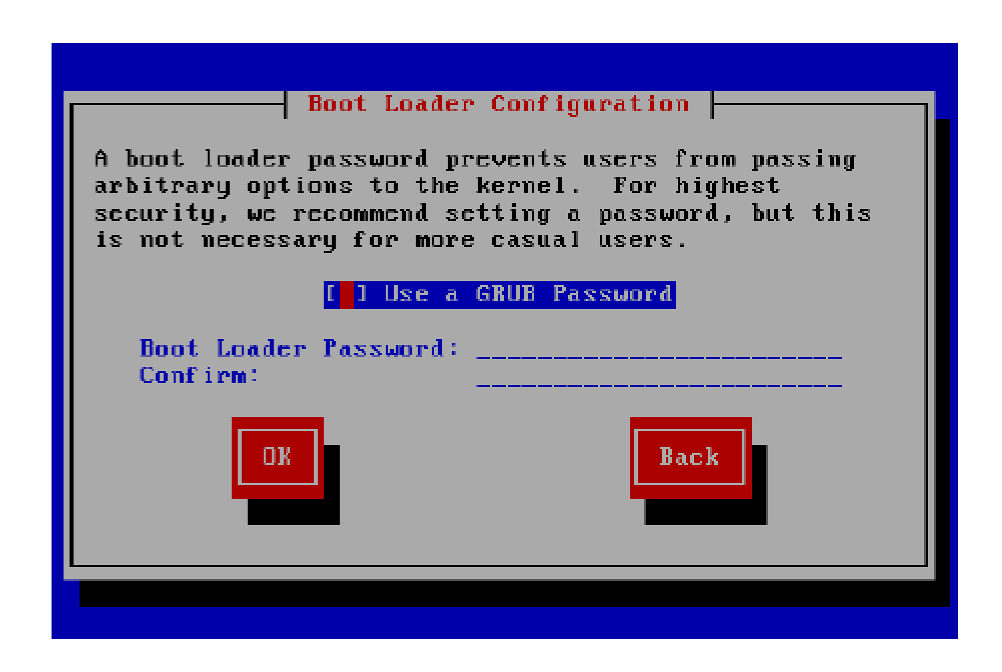


图 2-31　设置 GRUB 密码

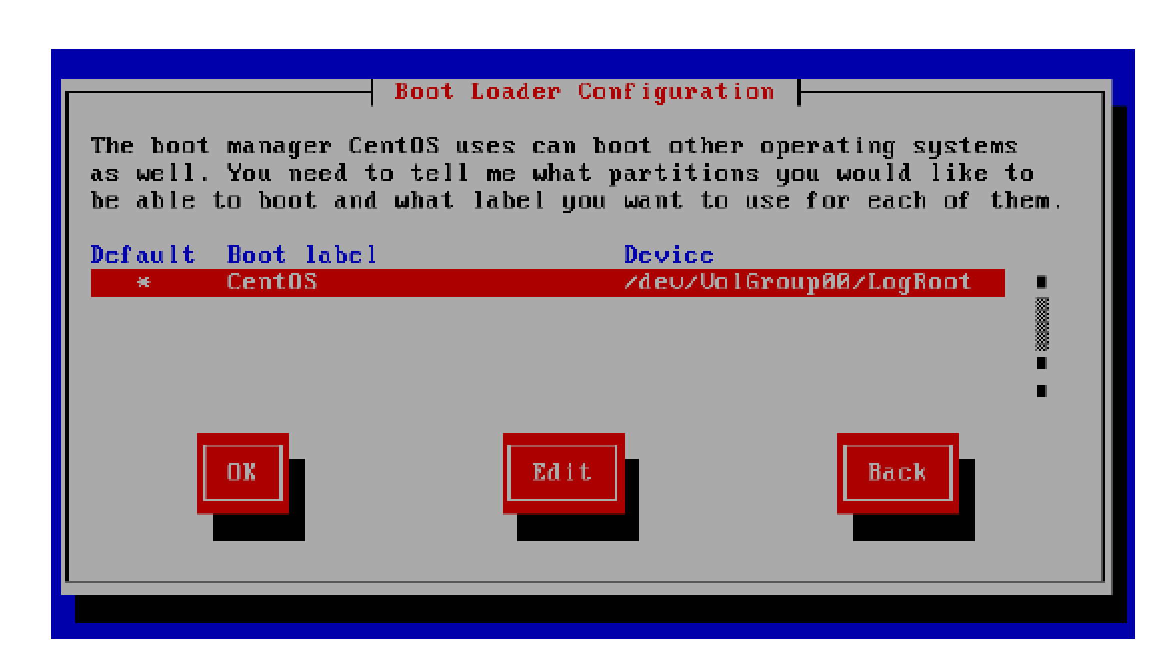


图 2-32　设置启动卷

(11) 进入启动加载器安装位置界面，如图 2-33 所示。使用默认值，选择 OK 按钮继续。

(12) 进入网络设置界面，如图 2-34 所示，选择 Yes 按钮开始进行网络设置。

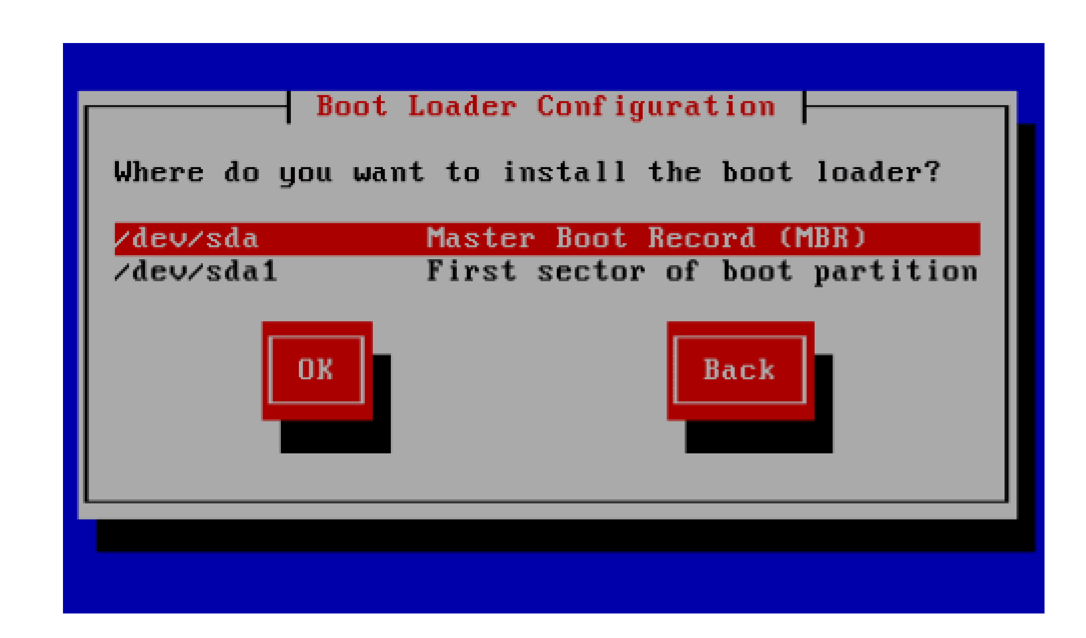
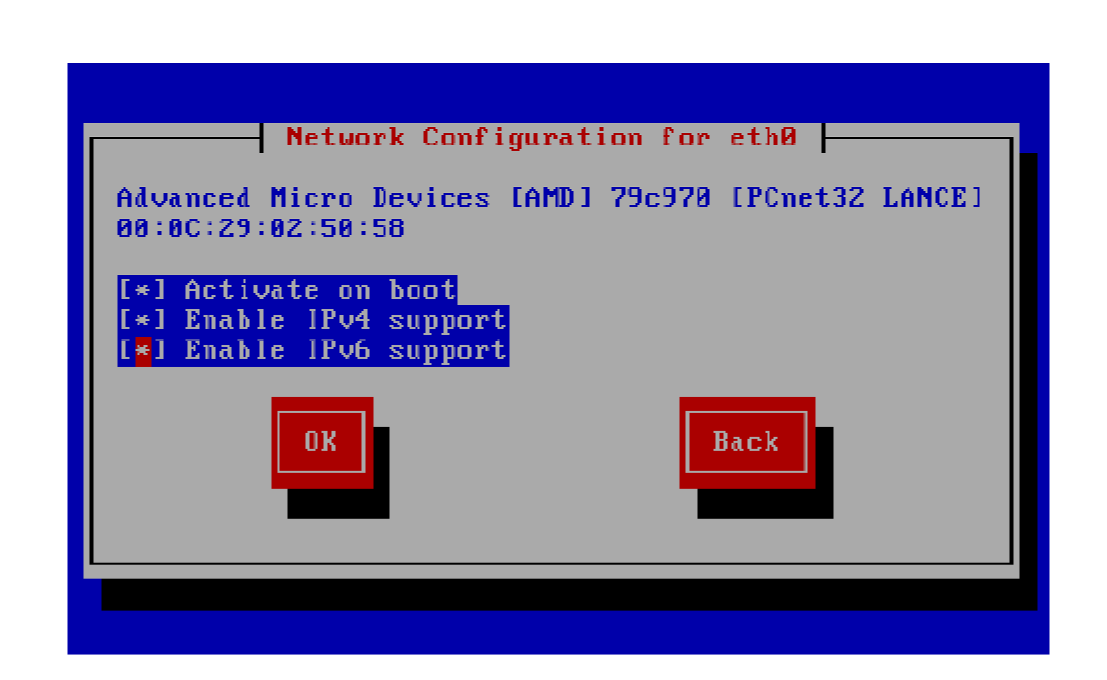


图 2-33　设置启动加载器安装位置

图 2-34　开始设置网络

(13) 进入选择网络协议界面，如图 2-35 所示，开启网卡对 IPv4 和 IPv6 的支持选项，选择 OK 按钮继续。

(14) 进入网络地址设置界面，如图 2-36 所示。在绝大多数情况下，服务器是使用静态 IP 地址的，选择手动地址设置选项，设置 IP 地址及子网掩码，选择 OK 按钮继续。



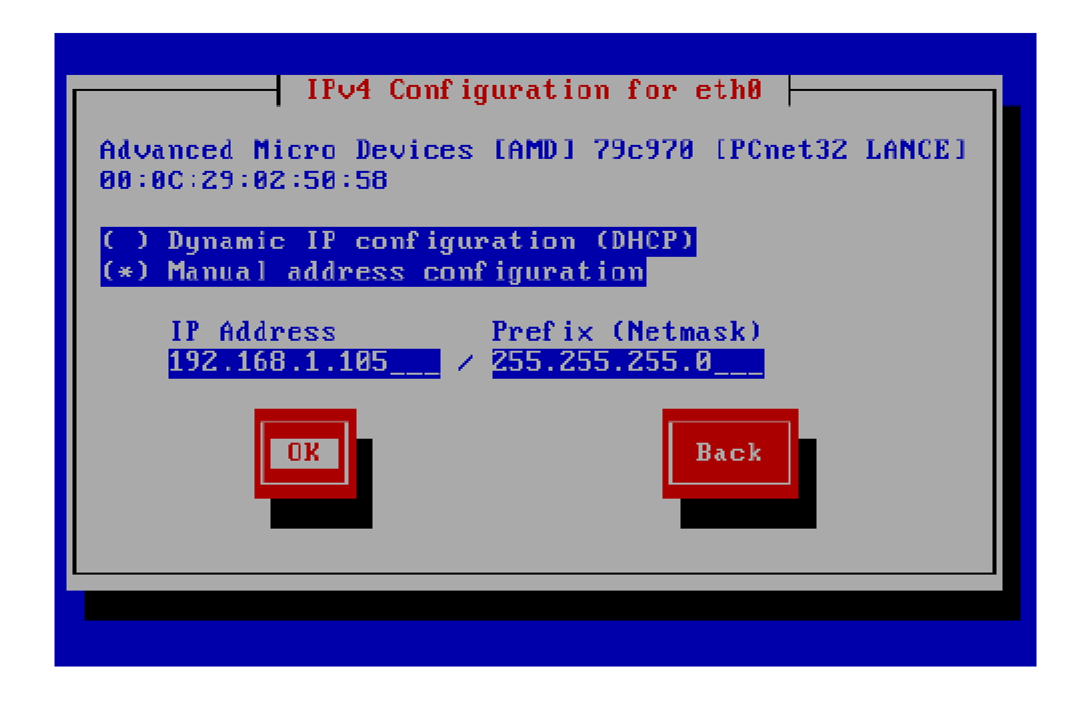
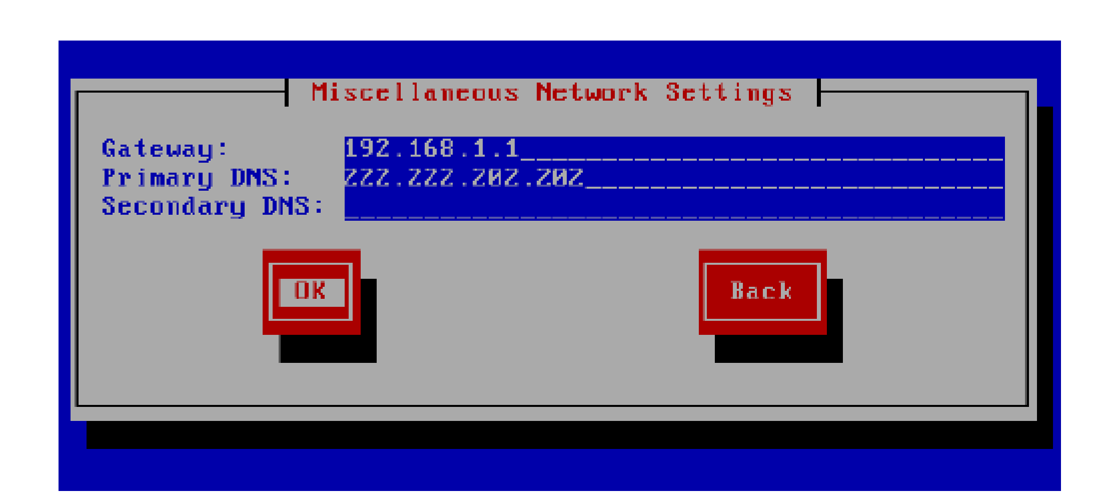


图 2-35　开启网络协议

图 2-36　设置 IP 地址

(15) 进入设置默认网关和 DNS 界面，如图 2-37 所示。填写服务器的默认网关和 DNS 服务器地址，选择 OK 按钮继续。

(16) 进入主机名获取方式界面如图 2-38 所示，如果服务器所在网络的 DHCP 服务器开启了分发主机名的功能，可以选择通过 DHCP 自动获得选项，否则，选择手动设置选项，并输入主机名。选择 OK 按钮继续。



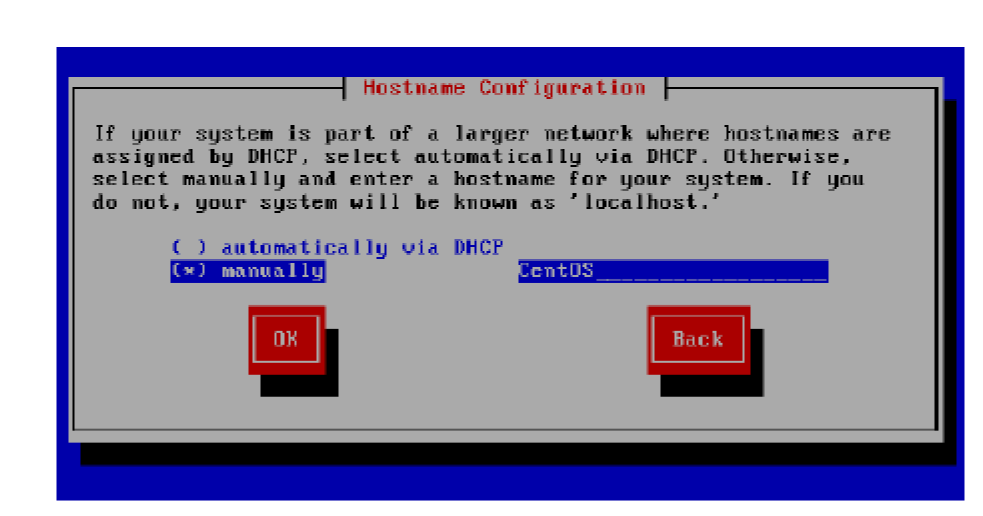


图 2-37　设置默认网关和 DNS 服务器

图 2-38　设置主机名

(17) 进入选择时区界面，如图 2-39 所示。选择 Asia/Shanghai 时区，选择 OK 按钮继续。

(18) 进入管理员密码设置界面，如图 2-40 所示。输入 Root 用户的密码，选择 OK 按钮继续。

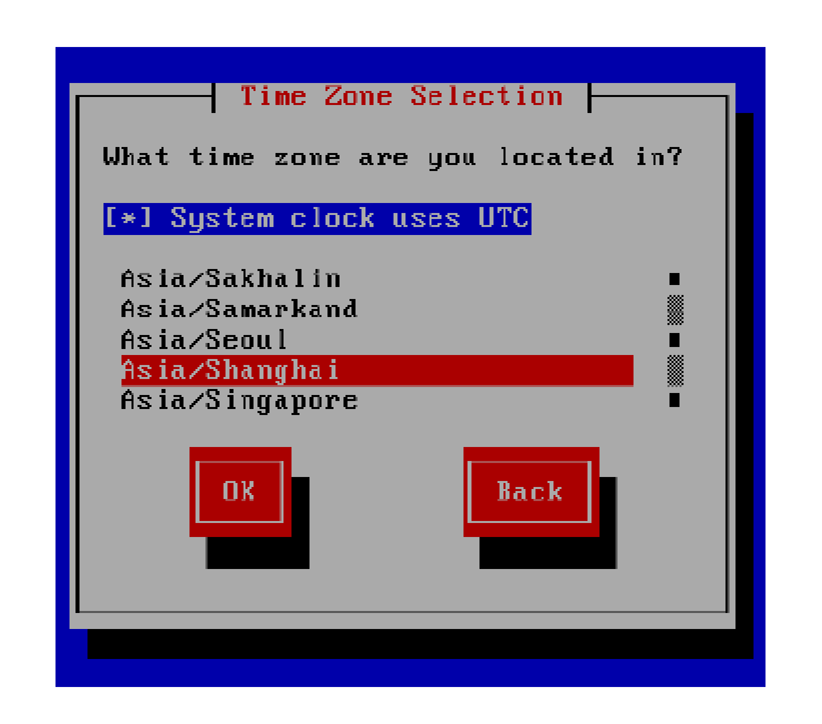


图 2-39　选择时区

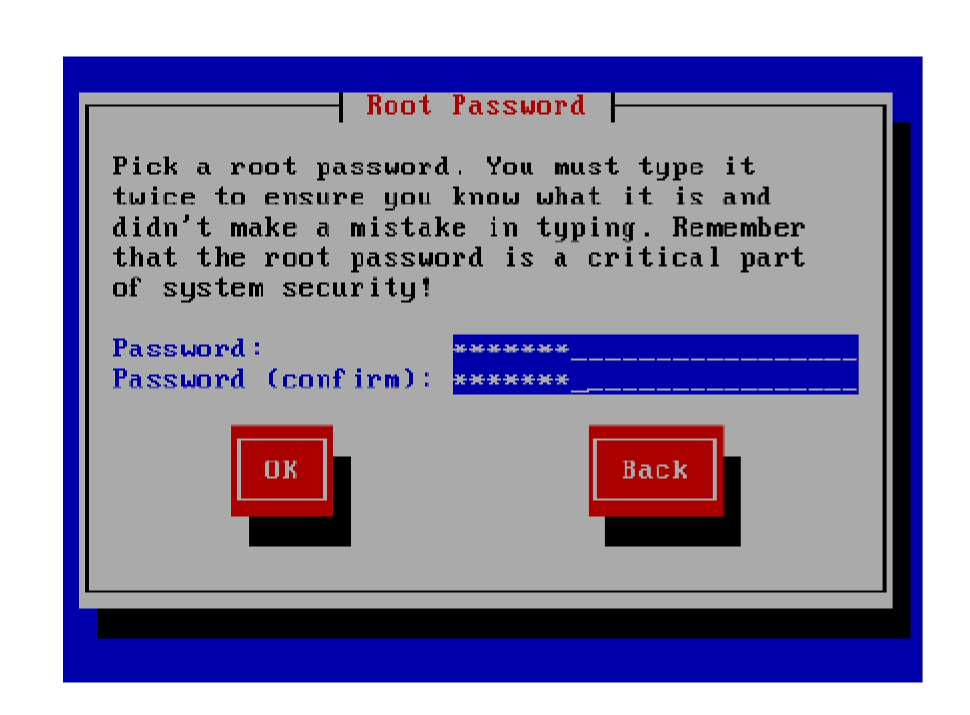


图 2-40　输入 root 密码

(19) 进入安装包选择界面，如图 2-41 所示。选择服务器的使用类型后，安装程序会根据不同类型的服务器选择不同的安装包组件，此处我们选择 Server-GUI 服务器类型，选择 OK 按钮继续。

(20) 进入开始安装界面，如图 2-42 所示。管理员可以最后确认是否开始安装，选择 OK 按钮，开始安装 CentOS。



图 2-41　选择安装包

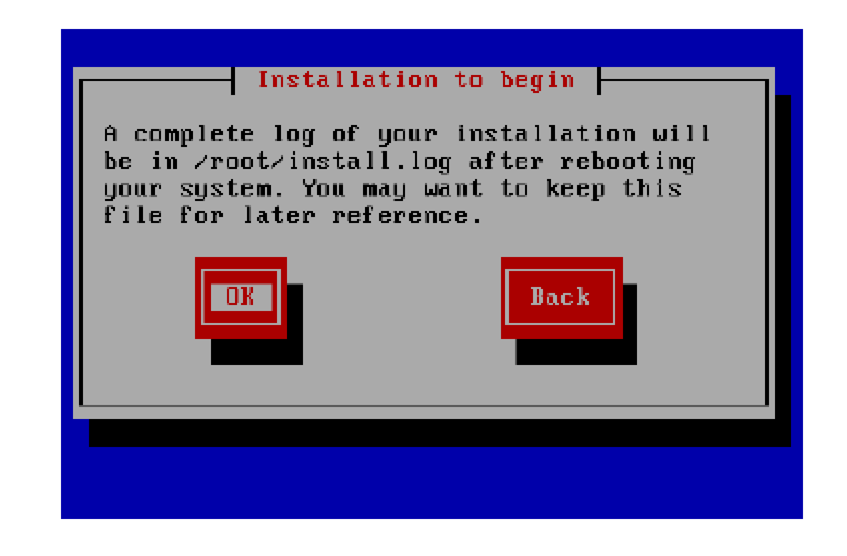


图 2-42　确认安装

(21) 开始安装 CentOS 操作系统，安装过程如图 2-43 所示。

(22) 安装完成后，安装程序会弹出对话框，要求重启服务器，如图 2-44 所示，这也标志着我们的 Linux 已经安装成功。

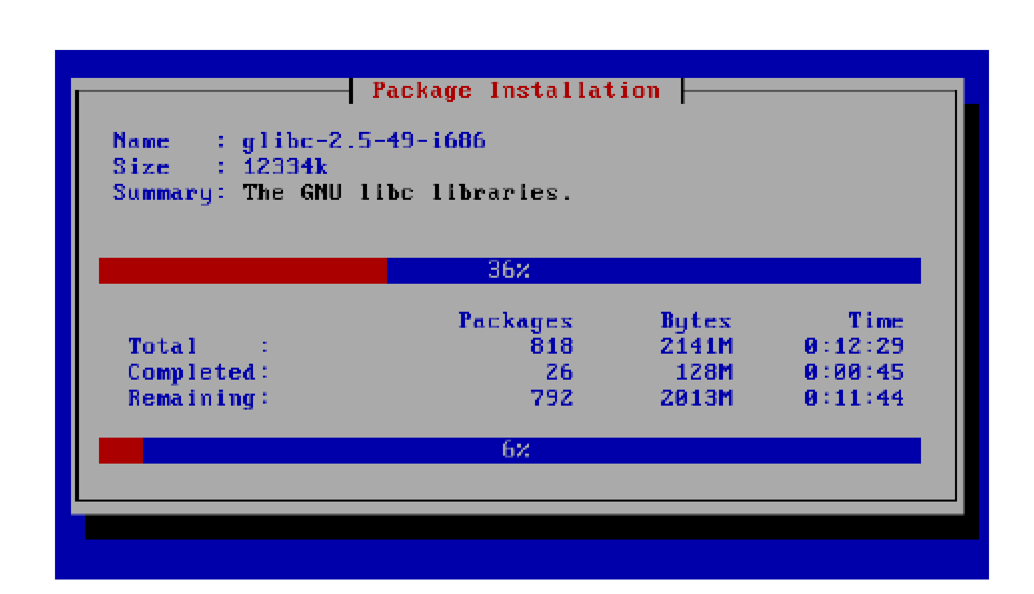


图 2-43　安装过程



图 2-44　重启服务器

2.3.3 使用 Setup Agent

当 CentOS 安装完成后，第一次启动操作系统时会首先启用 Setup Agent，如图 2-45 所示。

Setup Agent 可以对我们在安装 Linux 过程中的一些内容进行再一次的设置，如时区、键盘等设置。例如，我们以修改防火墙设置，那么可以选择 FireWall configuration 选项，然后选择 Run Tool 按钮，我们可以在防火墙设置中打开或者关闭系统防火墙，如图 2-46 所示。







图 2-45 Setup Agent 界面　　　　图 2-46 设置防火墙

提示： *在实际应用中，管理员一般都会自己安装防火墙，而不是使用 Linux 内置的防火墙。这时，就需要将系统防火墙禁止。*

2.3.4 使用 yum 工具

在 Linux 的系统管理中，最繁琐的问题要数软件包之间的依赖问题了，由于 Linux 发布版本众多，对应不同版本、不同核心的软件也是五花八门，经常会产生各种各样的版本问题，往往是明明已经安装了依赖软件，但因为版本不同，也不能正常使用。

为了解决上述问题，开源社区开发了多种解决发行版本依赖关系的工具，在 Red Hat 中最常用的要数 yum 了。yum(全称为 Yellow dog Updater, Modified)是在 Fedora、Red Hat 以及 SUSE、CentOS 中的 Shell 前端软件包管理器。基于 RPM 包管理，能够从指定的服务器自动下载 RPM 包并且安装，可以自动处理依赖性关系，并且一次安装所有依赖的软件包，无须繁琐地一次次下载、安装。目前，Red Hat 和 CentOS 等 Linux 系统都已经默认安装 yum 软件。

要正常使用 yum，首先需要修改其更新源，目前国内访问速度较快的常用更新网址有以下几个：

- http://ftp.sjtu.edu.cn/centos/
- http://centos.candishosting.com.cn/
- http://ftp.hostrino.com/pub/centos/
- http://mirrors.ta139.com/centos

可以通过编辑 yum 的配置文件 CentOS-Base.repo 来将这些更新源添加到 yum 中。打

开并且编辑文本文件的命令为：

```
#vi /etc/yum.repos.d/CentOS-Base.repo
```

我们需要将以上的 4 个更新源分别添加到[base]、[updates]、[addons]、[extras]、[centosplus]和[contrib]的相应位置，修改后的 CentOS-Base.repo 文件内容如下面的代码所示：

```
# CentOS-Base.repo
#
# The mirror system uses the connecting IP address of the client and the
# update status of each mirror to pick mirrors that are updated to and
# geographically close to the client.  You should use this for CentOS
updates
# unless you are manually picking other mirrors.
#
# If the mirrorlist= does not work for you, as a fall back you can try
the
# remarked out baseurl= line instead.
#
#

[base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basea
rch&repo=os
#baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
http://ftp.sjtu.edu.cn/centos/$basearch/
http://centos.candishosting.com.cn/$basearch/
http://ftp.hostrino.com/pub/centos/$basearch/
http://mirrors.ta139.com/centos/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5

#released updates
[updates]
name=CentOS-$releasever - Updates
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basea
rch&repo=updates
#baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
http://ftp.sjtu.edu.cn/centos/$basearch/
http://centos.candishosting.com.cn/$basearch/
http://ftp.hostrino.com/pub/centos/$basearch/
http://mirrors.ta139.com/centos/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5

#packages used/produced in the build but not released
[addons]
name=CentOS-$releasever - Addons
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basea
rch&repo=addons
#baseurl=http://mirror.centos.org/centos/$releasever/addons/$basearch/
http://ftp.sjtu.edu.cn/centos/$basearch/
http://centos.candishosting.com.cn/$basearch/
http://ftp.hostrino.com/pub/centos/$basearch/
http://mirrors.ta139.com/centos/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5
```

```
#additional packages that may be useful
[extras]
name=CentOS-$releasever - Extras
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basea
rch&repo=extras
#baseurl=http://mirror.centos.org/centos/$releasever/extras/$basearch/
http://ftp.sjtu.edu.cn/centos/$basearch/
http://centos.candishosting.com.cn/$basearch/
http://ftp.hostrino.com/pub/centos/$basearch/
http://mirrors.ta139.com/centos/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5

#additional packages that extend functionality of existing packages
[centosplus]
name=CentOS-$releasever - Plus
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basea
rch&repo=centosplus
#baseurl=http://mirror.centos.org/centos/$releasever/centosplus/$basearc
h/
http://ftp.sjtu.edu.cn/centos/$basearch/
http://centos.candishosting.com.cn/$basearch/
http://ftp.hostrino.com/pub/centos/$basearch/
http://mirrors.ta139.com/centos/$basearch/
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5

#contrib - packages by Centos Users
[contrib]
name=CentOS-$releasever - Contrib
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basea
rch&repo=contrib
#baseurl=http://mirror.centos.org/centos/$releasever/contrib/$basearch/
http://ftp.sjtu.edu.cn/centos/$basearch/
http://centos.candishosting.com.cn/$basearch/
http://ftp.hostrino.com/pub/centos/$basearch/
http://mirrors.ta139.com/centos/$basearch/
gpgcheck=1
enabled=0
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5
```

编辑完成后，我们就可以使用 yum 来更新系统软件了。更新系统的命令如下：

```
#rpm -import /etc/pki/rpm-gpg/RPM-GPG-KEY*
#yum upgrade
```

2.4 GNOME 桌面环境使用与管理

对于用惯了 Windows 系统的用户来说，对于 Linux 中传统的命令行形式比较难以接受，更喜欢 Windows 中的图形化用户界面环境(GUI)，而对于使用 Linux 作为娱乐和办公的用户来说，图形化用户界面环境更加友好，为了响应这些需求，Linux 操作系统其实很早就已经开始提供这种图形化的用户界面解决方案。

2.4.1 X Window 简介

X Window 系统(X Window System，也常称为 X11 或 X)是一种以位图方式显示的软件窗口系统。最初是 1984 年麻省理工学院研究的，之后变成 UNIX、类 UNIX 以及 OpenVMS 等操作系统所一致使用的标准化软件工具包及显示架构的运作协议。X Window 系统通过软件工具及架构协议来建立操作系统所用的图形用户界面，此后则逐渐扩展适用到各形各色的其他操作系统上。现在几乎所有的 Linux 操作系统都能支持与使用 X Window。

X Window 系统主要有以下几个特征：

(1) X Window 系统具有网络透明性(Network Transparent)。透过网络，应用程序窗口在其他实际中的输出显示就和在自己机器上一样容易。

(2) 可以支持多种不同风格的操作系界面。X Window 系统只是提供建立窗口的一个标准，而管理窗口的功能(例如，窗口的摆放、大小以及显示顺序等)并不包含在系统中，而是由应用程序来控制，因此可以轻易地切换。

(3) X Window 系统不是操作系统的一部分，对于操作系统而言，X Window 只是一个程序而已，因此很容易在不同的系统上安装。

(4) X Window 系统的窗口是层次性的。应用程序可以直接使用窗口系统已有的设施便可满足大部分的需求，而无需借助其他的输入或者控制结构。

(5) X Window 系统是免费的。X Window 系统是开源项目，可通过网络或者其他途径免费获得源代码。

更重要的是，今日知名的桌面环境——GNOME 和 KDE 也都是以 X Window 系统为基础建构成的。

2.4.2 GNOME 桌面环境介绍

GNOME(GNU Network Object Model Environment，GNU 网络对象模型环境)最初是由墨西哥的程序设计师发起的，为了克服 KDE 桌面环境所遇到的许可协议和对 C++依赖的问题，它受到了 Red Hat 公司的大力支持，其主要目的是希望能够为用户提供一个完整易学易用的桌面环境，并为程序设计师提供强大的应用程序开发环境。

2.4.3 GNOME 桌面环境的使用

熟悉 Windows 操作系统的管理员对于 GNOME 桌面上手会更快一些，因为 GNOME 的操作方式与 Windows 操作系统非常相似。

1. 进入 GNOME

启动 CentOS 操作系统后，默认是以命令行界面，如果要进入图形界面，可以输入以下命令：

```
[root@CentOS ~]# #startx
```

经过短暂的载入，我们可以看到如图 2-47 所示的图形界面，这就是 GNOME 的默认界面环境。在 CentOS 中，默认使用的是 GNOME 桌面环境，它主要由以下几个部分构成。

- 桌面以及桌面上的图标、窗口。
- 面板位于桌面的上方和下方。
- 任务栏位于面板中。

如果用户希望切换到其他桌面环境，可以在“终端(Terminal)”中输入以下命令：

```
[root@CentOS ~]# switchdesk
```

会弹出如图 2-48 所示的“桌面切换器”对话框，管理员可以在其中选择自己喜欢的桌面环境。

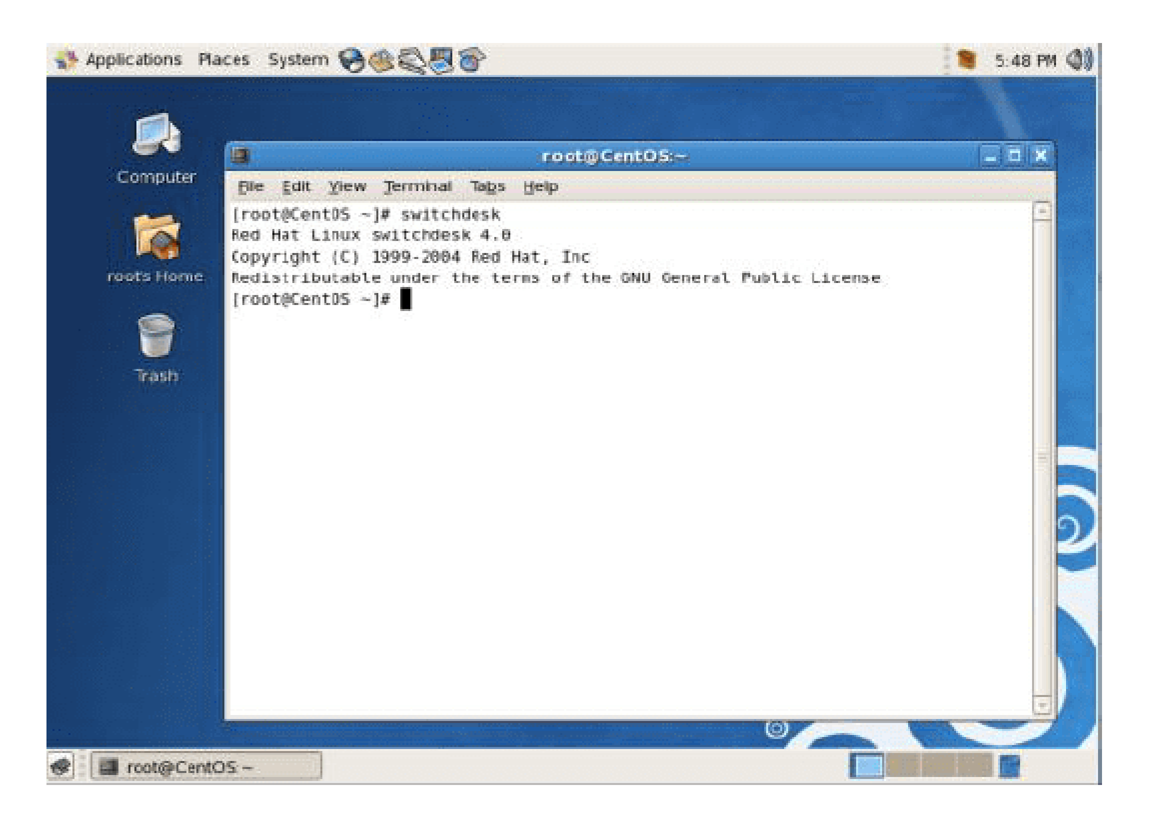


图 2-47 GNOME 默认桌面环境

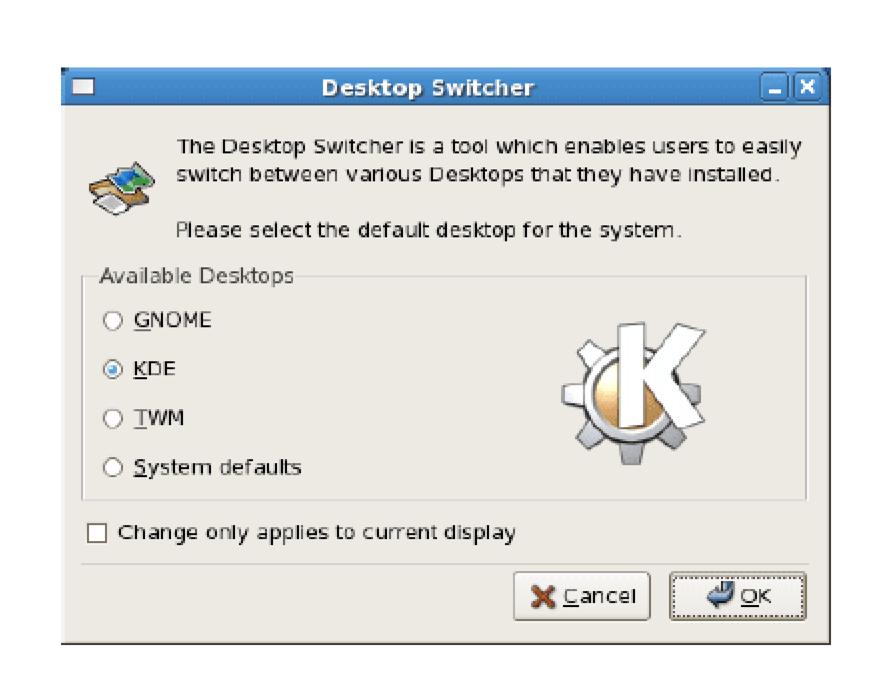


图 2-48 桌面切换器对话框

2. 使用 GNOME 面板

GNOME 面板是 GNOME 桌面的重要组成部分，位于桌面的上方和下方，在面板中集成了很多的常用工具。默认的面板构成如图 2-49 所示。

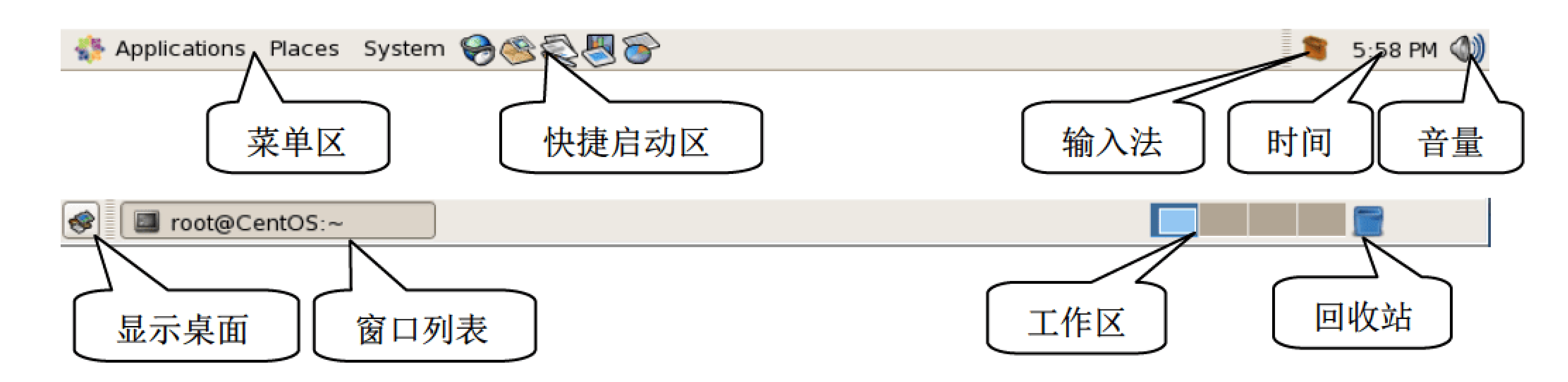


图 2-49 面板构成

与 Windows 的任务栏类似，GNOME 的面板也是可以自定义的，管理员可以自定义面板的属性、添加程序到快速启动区以及定制虚拟桌面等。

1) 自定义面板属性

自定义面板属性的操作包括设置面板方式的位置与大小等，具体的操作步骤如下。

(1) 在面板的空白处单击鼠标右键，在弹出的快捷菜单中选择 properties(属性)命令，如图 2-50 所示。

(2) 弹出 Panel Properties(面板属性)对话框如图 2-51 所示，如果需要设置面板的位置，可以在 Genernal(常规)标签中的 Orientation(方向)下拉列表框中选择，一共有 4 个选项，分别是 Top(上)、Bottom (下)、Left(左)、Right(右)。

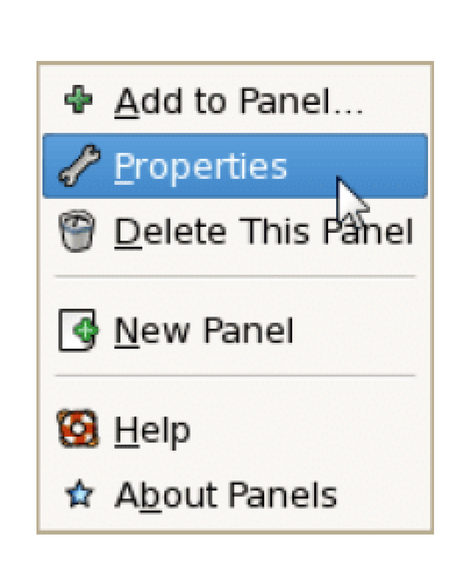


图 2-50　选择 Properties(属性)命令



图 2-51　Panel Properties(面板属性)对话框

(3) 如果要设置面板的大小，可以在 Size(大小)文本框中进行设置。

(4) 如果需要自动隐藏面板，可以选中 Autohide(自动隐藏)复选框。如果管理员希望自己手动隐藏面板，也可以选中 Show hide buttons(显示隐藏按钮)复选框。

(5) 切换到 Background(背景)选项卡，可以设置面板的背景图片或者是颜色，如图 2-52 所示。

(6) 设置完成后，单击 Close(关闭)按钮可以关闭此对话框。

2) 在快速启动区中添加程序

在菜单区右侧的快速启动区中，通常放置着经常使用的应用程序的启动命令，管理员可以通过单击某个项目启动相应的程序，管理员也可以将自己的应用程序添加到快速启动区中。

在快速启动区中添加应用程序的具体操作步骤如下。

(1) 在面板空白处单击鼠标右键，从弹出的快捷菜单中选择 Add to Panel(添加到面板)命令，如图 2-53 所示。

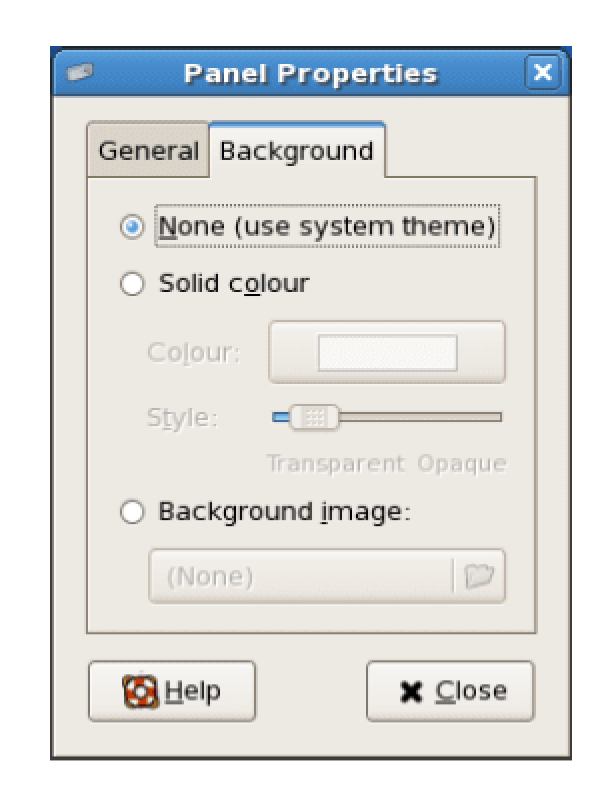


图 2-52　Backgroud(背景)选项卡

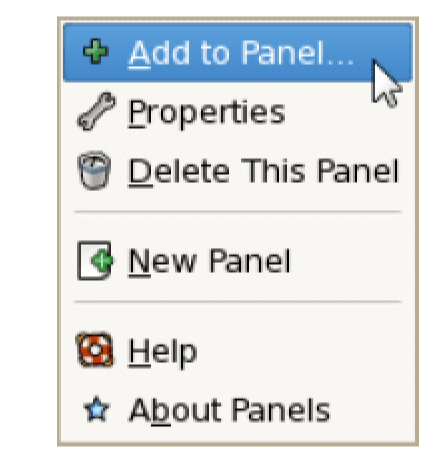


图 2-53　Add to Panel(添加到面板)命令

(2) 弹出 Add to Panel(添加到面板)对话框，如图 2-54 所示。管理员可以在列表中选择系统自带的程序(例如选择 Dictionary Look up(字典)程序)，然后单击 Add(添加)按钮将此程

序添加到面板中，添加后的样式如图 2-55 所示。



图 2-54 Add to Panel(添加到面板)对话框

(3) 在创建好的应用程序启动器上单击鼠标右键，在弹出的快捷菜单中选择 Move(移动)命令可以将新添加的应用程序移动到需要的位置，移动后的效果如图 2-56 所示。

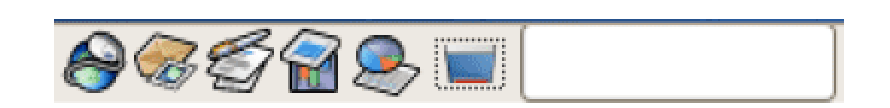


图 2-55 添加后的应用程序

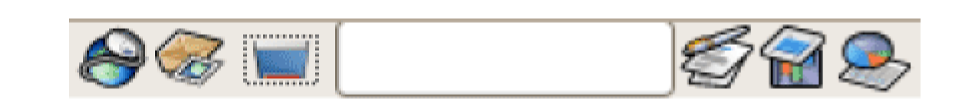


图 2-56 移动后的应用程序

(4) 位置调整完成后，在快捷菜单中选择 Lock to Panel(锁定到面板)命令，如图 2-57 所示，可以防止启动器再次被移动。

如果在 Add to Panel(添加到面板)对话框中没有需要的应用程序，管理员也可以选择 Custom Application Launcher(自定义应用程序)命令来手动添加需要的应用程序。下面我们就以添加一个“关机”启动器为例，介绍手动添加应用程序的具体操作步骤。

(1) 在面板空白处单击鼠标右键，从弹出的快捷菜单中选择 Add to Panel(添加到面板)命令。

(2) 弹出 Add to Panel(添加到面板)对话框，选择 Custom Application Launcher(创建应用程序启动器)选项，再单击 Add(添加)按钮。

(3) 弹出 Create Launcher(创建启动器)对话框，如图 2-58 所示。在 Name(启动器名称)文本框中输入命名的启动器名称 poweroff，再单击 Command(命令)右侧的 Browse(浏览)按钮。

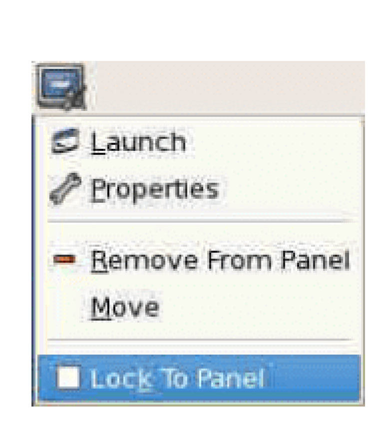


图 2-57 选择 Lock to Panel(锁定到面板)命令

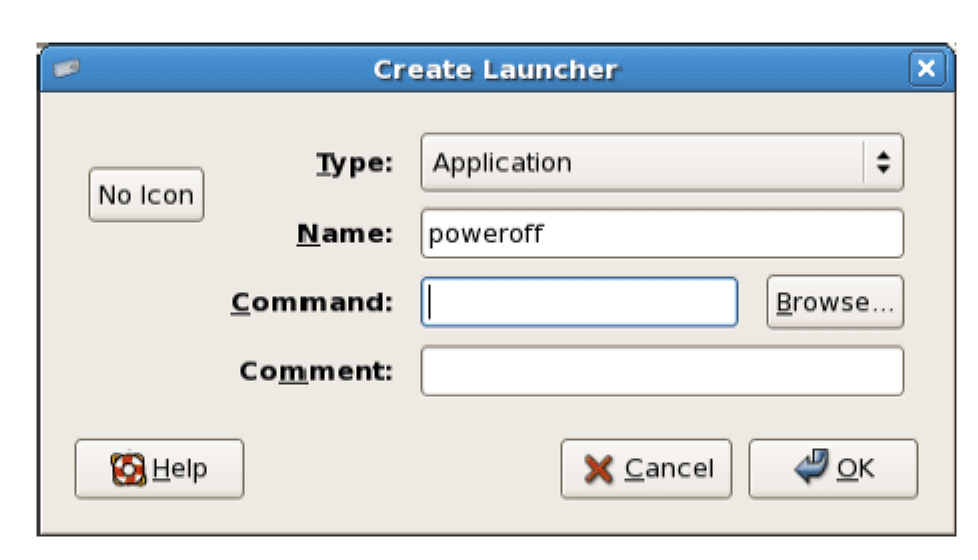


图 2-58 Create Launcher(创建启动器)对话框

(4) 弹出 Choose an application(选择应用程序)对话框，如图 2-59 所示。依次选择 File System→sbin→poweroff 选项，再单击 Open(打开)按钮完成命令的添加。返回 Create Launcher(创建启动器)对话框。

(5) 另外，如果需要，管理员还可以单击对话框中的 No Icon 按钮来添加命令图标。如图 2-60 所示，弹出 Browse icons(浏览图标)对话框，在其中选择需要的命令图标即可。

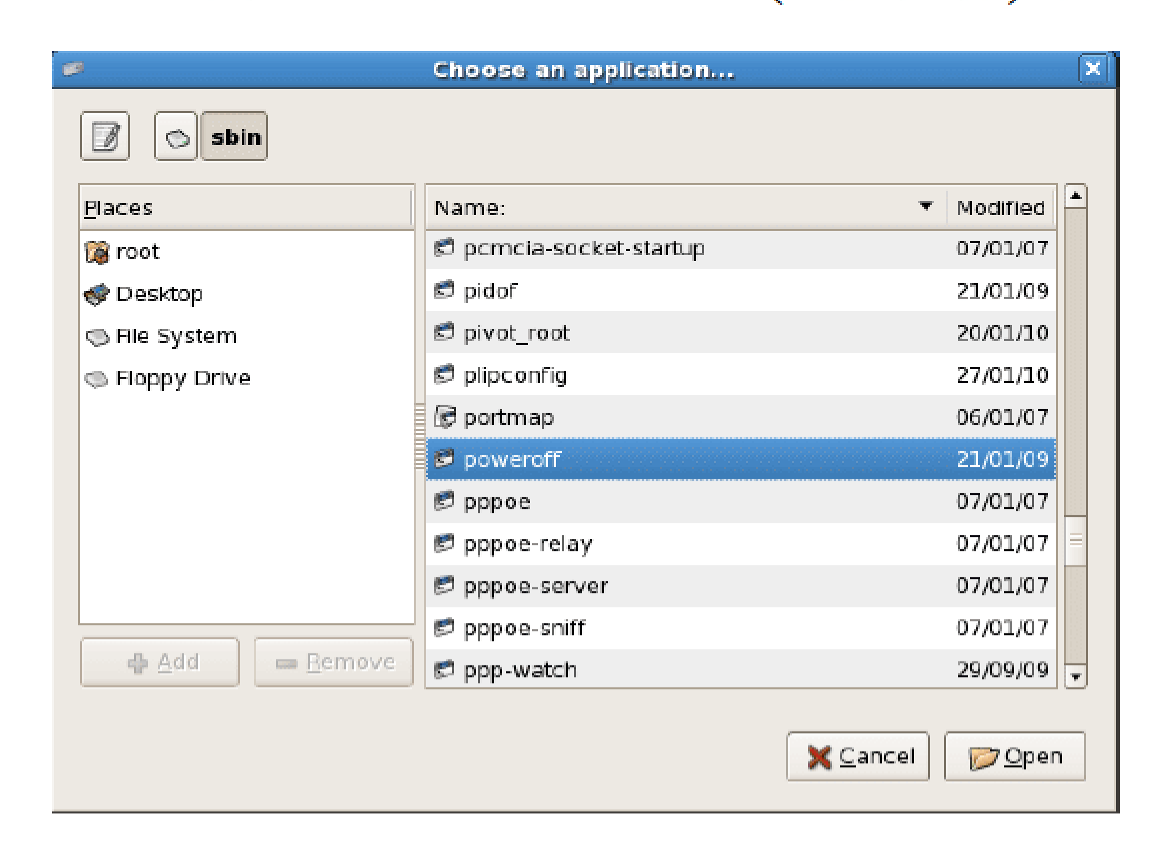


图 2-59　选择命令

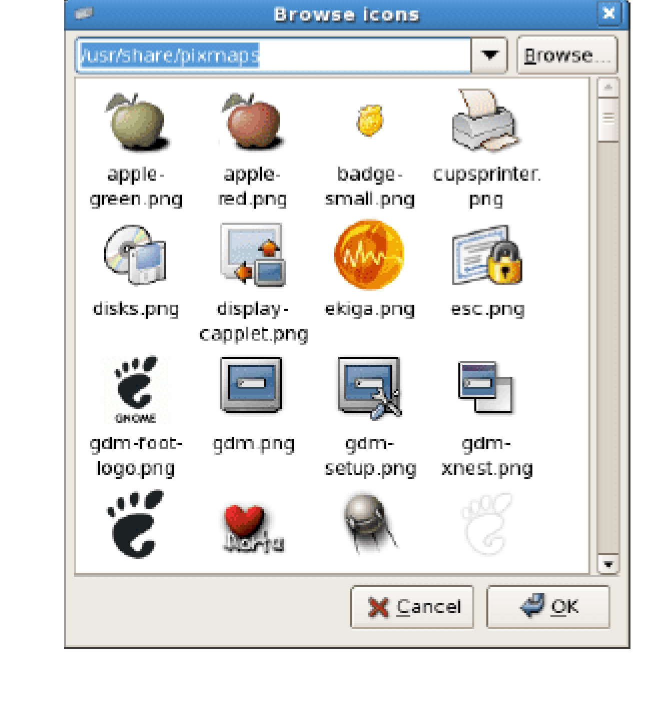


图 2-60　选择图标

(6) 命令配置完毕后，如图 2-61 所示，单击 OK 按钮即可，此时命令会显示在面板中，如图 2-62 所示，说明已经成功添加 poweroff(关机命令)启动器。

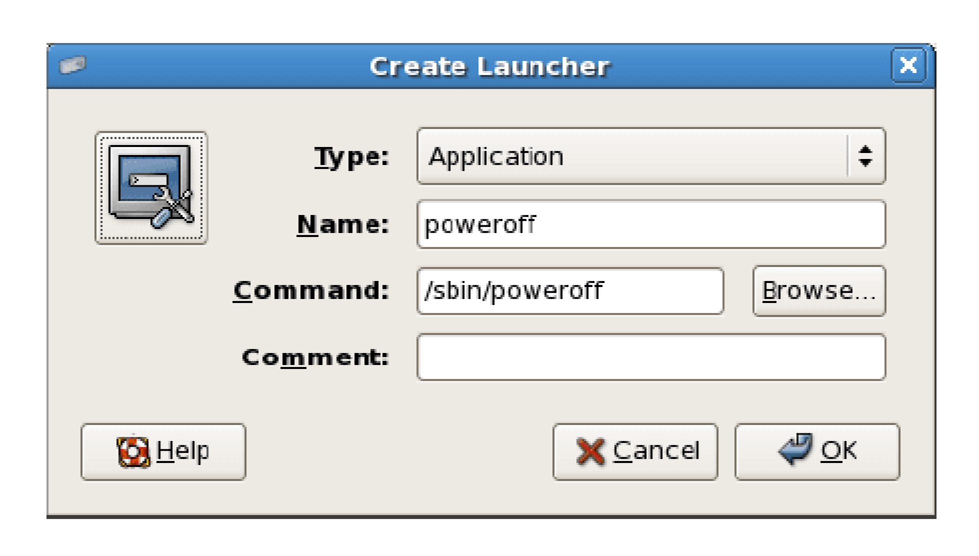


图 2-61　配置完成的创建启动器对话框

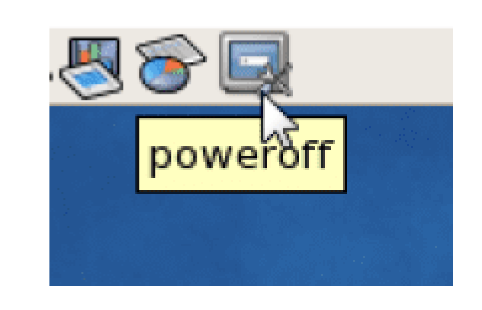


图 2-62　成功添加 poweroff 启动器

3) 定制工作区

在 GNOME 的面板右下角，有一个工作区(在有些资料中也称为虚拟桌面)区域如图 2-49 所示，它可以非常方便的管理应用程序窗口。在 GNOME 中默认一共有 4 个工作区，管理员可以将打开的窗口放在不同的工作区中，通过切换工作区来访问不同的窗口。例如，把浏览器窗口放在“工作区 1”中，而把 OpenOffice 窗口放在“工作区 2”中。工作区之间的切换主要有以下两种操作方法。

- 使用鼠标直接单击需要切换到的工作区。
- 使用快捷键 Ctrl+Alt+Left(Right)切换到左(右)侧的工作区。

另外，在 GNOME 中默认提供 4 个工作区，管理员也可以自己定制工作区的数量，定制工作区数量的具体操作步骤如下。

(1) 在工作区区域单击鼠标右键，在弹出的快捷菜单中选择 Preferences(首选项)命令，如图 2-63 所示。

(2) 弹出 Workspace Switcher Preferences(工作区切换器首选项)对话框，如图 2-64 所示。管理员在 Number of workspace(工作区数量)微调框中输入需要显示的工作区数量，单击 Close 按钮即可完成设置。

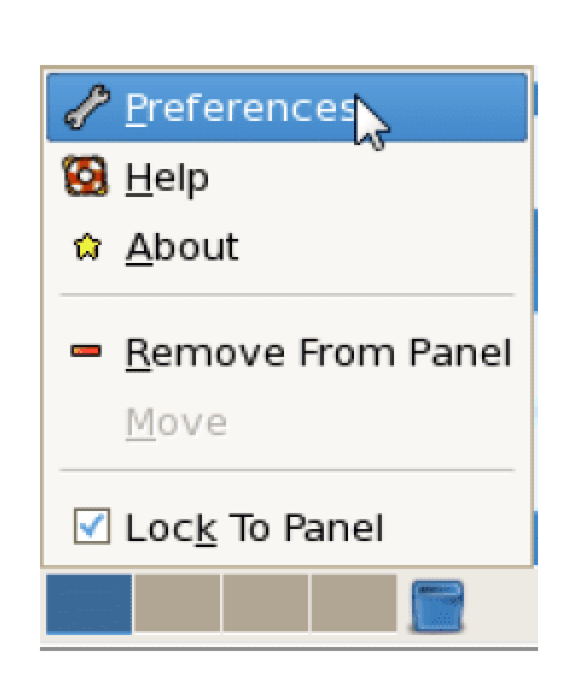


图 2-63 选择 Preferences(首选项)命令



图 2-64 Workspace Switcher Preferences (工作区切换器首选项)对话框

3. 设置窗口行为及主题

在 GNOME 桌面环境中，默认使用的窗口管理器是 metacity，通常不需要再更换其他的窗口管理器。如图 2-65 所示是 GNOME 桌面环境的默认典型窗口。



图 2-65 GNOME 桌面环境的典型窗口

一个典型的窗口应该包含标题栏、菜单栏、工作区、状态栏等几大元素。

实际上，GNOME 桌面环境中很多对窗口的操作方法与 Windows 操作系统非常类似，典型的操作方法包括但不限于以下几种：

- 双击标题栏可以最大化或者还原窗口大小。
- 如果窗口默认尺寸过大，超出了屏幕宽度或者高度，管理员不能看到整个窗口的所有内容。此时，可以按下 Alt 键，然后在窗口的任意位置按下鼠标左键进行拖动就可以移动窗口。

- 默认情况下，在一个工作区中打开的窗口不能在另一个工作区中出现，要调节窗口在工作区中出现的属性，可以右键单击此窗口的标题栏，弹出的快捷菜单如图 2-66 所示，选择相应的命令，可以实现对应的操作。例如，选择 Always on Visible Workspace(总在可见工作区)选项，当切换到其他工作区中时，此窗口也会一起切换过去；而选择 Move to Another Workspace(移动到其他工作区)选项，会弹出子菜单来，选择需要移动到的工作区，那么此窗口会切换到选定的工作区中。

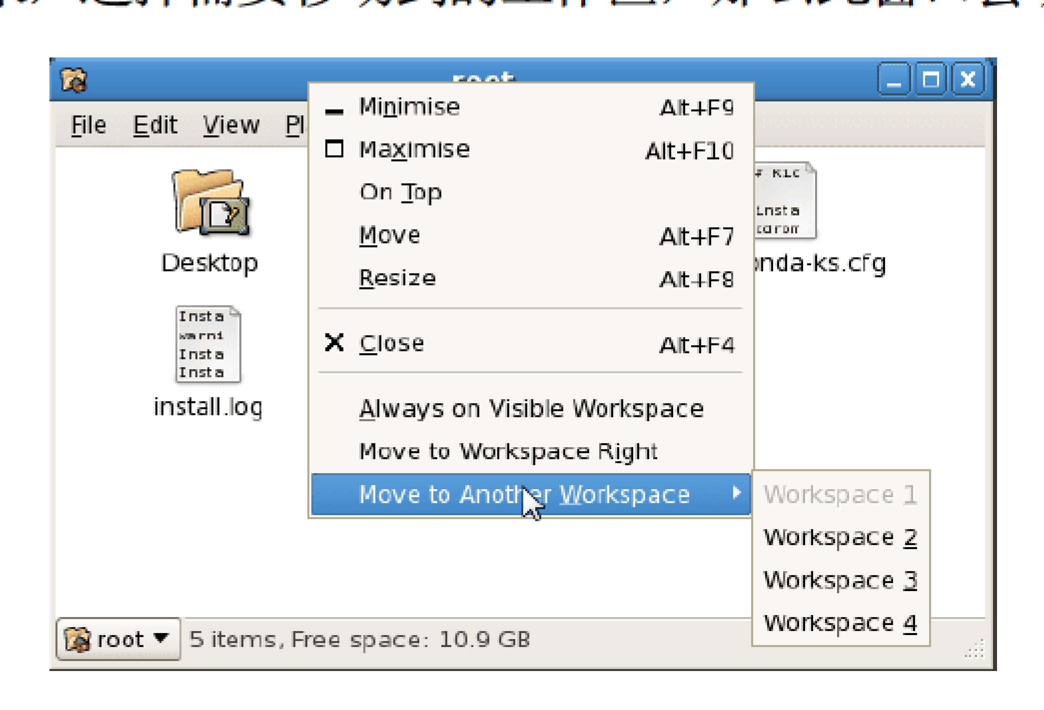


图 2-66　窗口标题栏快捷菜单

另外，GNOME 桌面环境中的窗口行为和主题都是可以设置的。设置窗口行为的具体操作步骤如下。

(1) 依次选择面板中的 System→Preferences→Windows 命令，如图 2-67 所示。

(2) 弹出 Window Preferences(窗口首选项)对话框，如图 2-68 所示。管理员可以对窗口选择、标题栏动作以及移动键等功能进行修改、设置，完成后单击 Close 按钮关闭对话框即可。

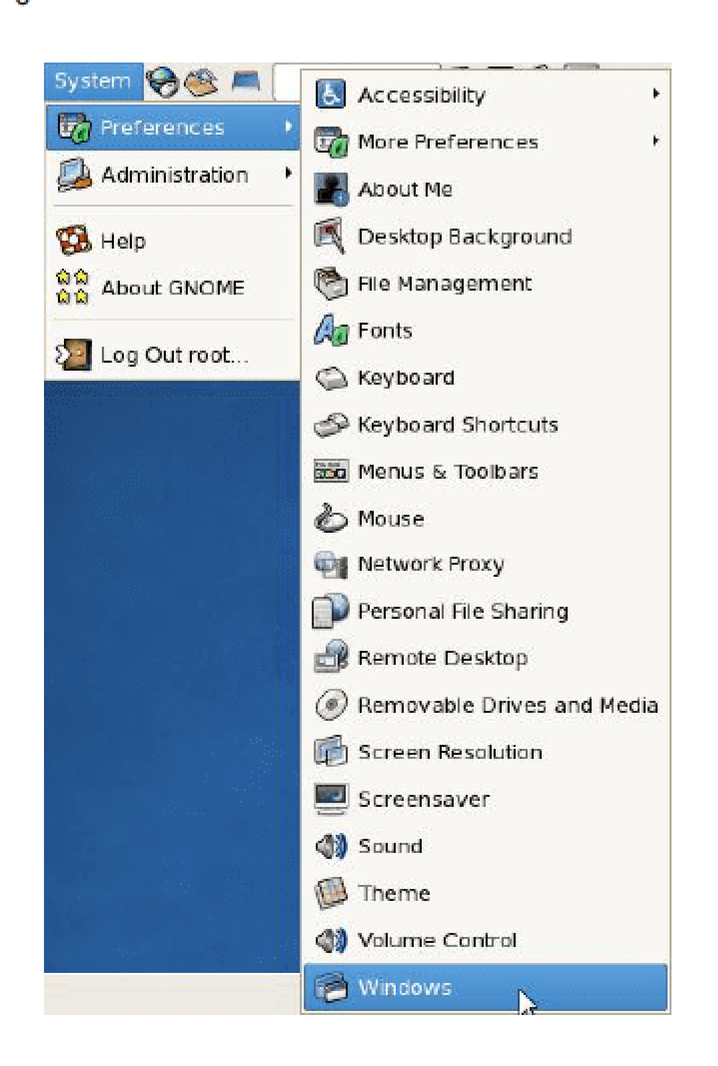




图 2-67　选择 Windows 命令



图 2-68　Window Preferences (窗口首选项)对话框

另外，GNOME 桌面环境还提供了丰富的窗口样式，用户可以根据自己的喜好，定制窗口的样式和各种窗口修饰元素的风格，这些内容统称为“窗口主题”。由于 GNOME 是开源的项目，所以如果用户喜欢，还可以在 Internet 中下载其他用户或者开发人员提供的主题包进行安装。GNOME 中定制窗口主题的具体操作步骤如下。

(1) 依次选择面板中的 System→Preferences→Theme 命令，如图 2-69 所示。

(2) 弹出 Theme Preferences(主题首选项)对话框，如图 2-70 所示，在其中选择某个主题后，可以看到该主题的显示效果，如果满意，可以单击 Close 按钮完成设置，如果不满意，也可以单击 Revert(还原)按钮恢复原来的设置。

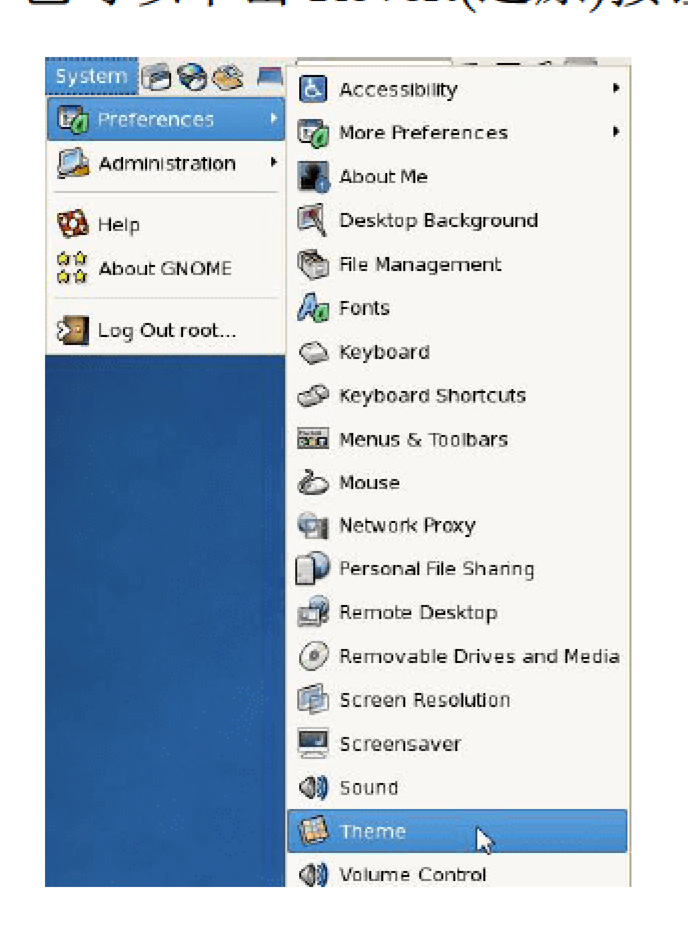


图 2-69 选择 Theme(主题)命令

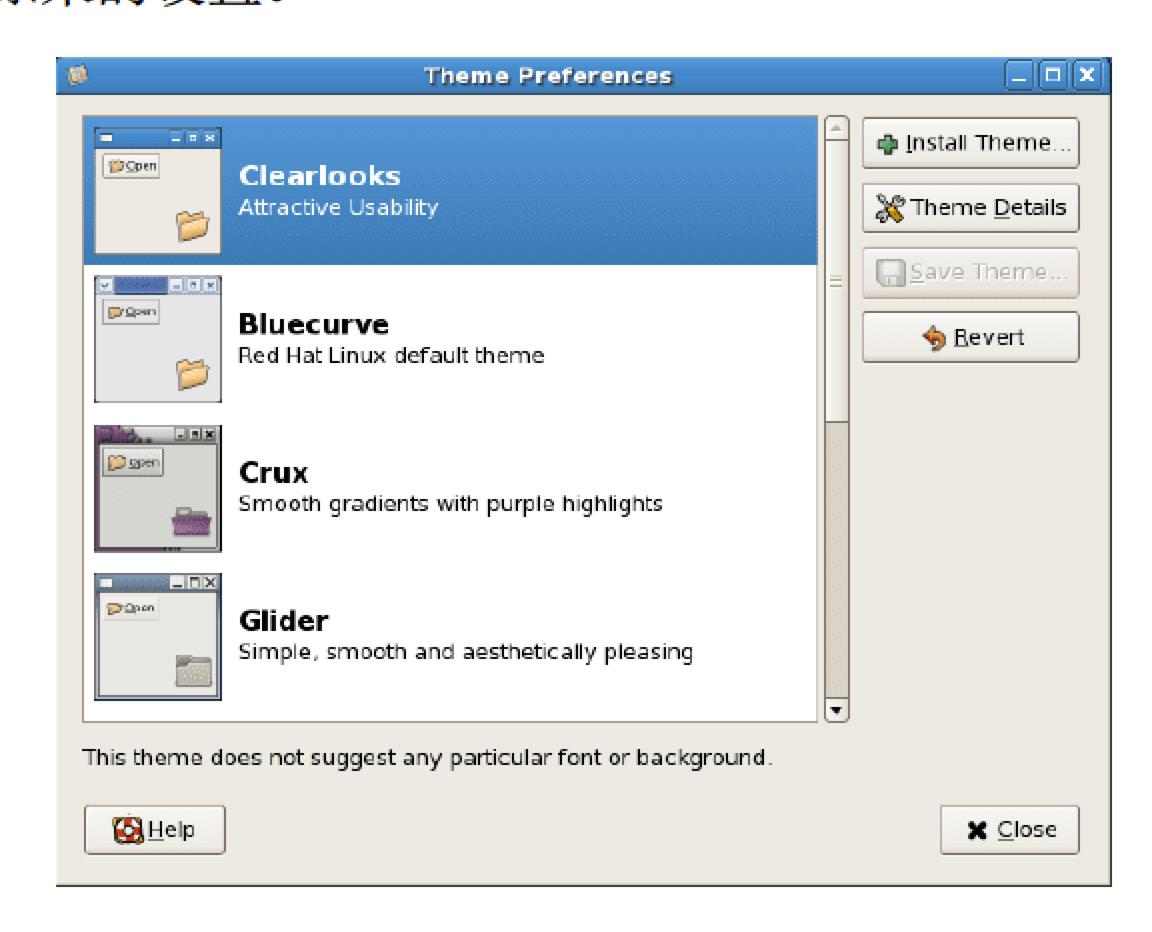


图 2-70 Theme Preferences(主题首选项)对话框

4. 设置文件管理器

GNOME 桌面环境中的文件管理器使用的是 Nautilus(鹦鹉螺)工具，打开文件管理器的方法是依次单击面板中的 Application→System Tools→File Browser 命令，如图 2-71 所示。除此之外，我们也可以通过双击桌面上的 root’s Home 图标来打开一个简化的文件管理器窗口。

默认的文件管理器窗口如图 2-72 所示。但其中的很多功能和显示效果都是可以进行自定义的，设置文件管理器窗口的方法是在文件管理器菜单中依次单击 Edit→Preferences 命令，如图 2-73 所示。弹出 File Management Preferences(文件管理器首选项)对话框，如图 2-74 所示，其中一共有 5 个选项卡，分别对应不同的设置功能。

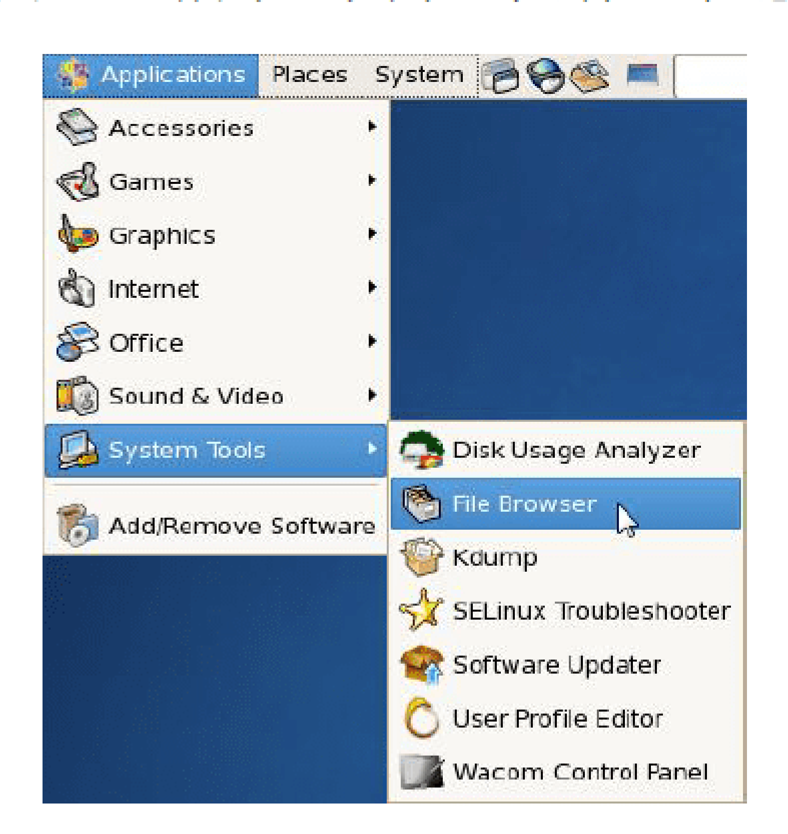


图 2-71 FileBrowser(文件管理器)命令

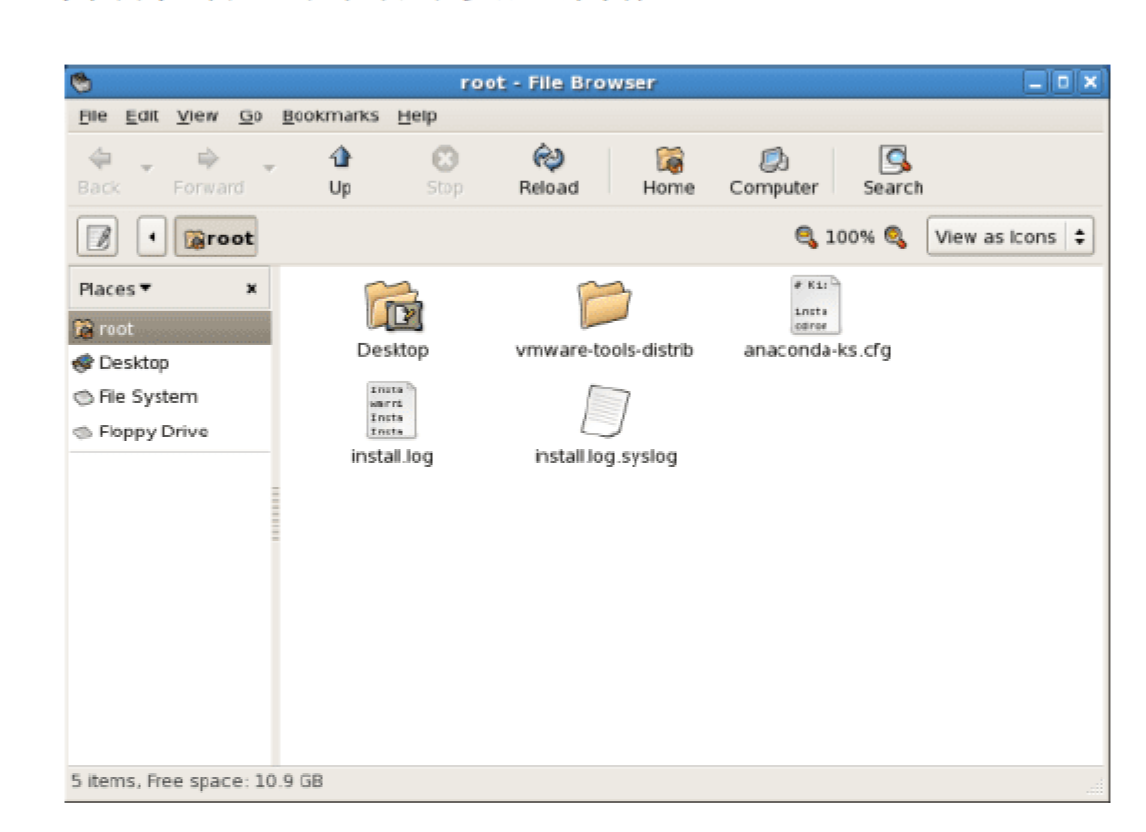


图 2-72 默认文件管理器界面

- Views(视图)选项卡：可以设置默认视图、图标大小、列表大小等功能，如图 2-74 所示。

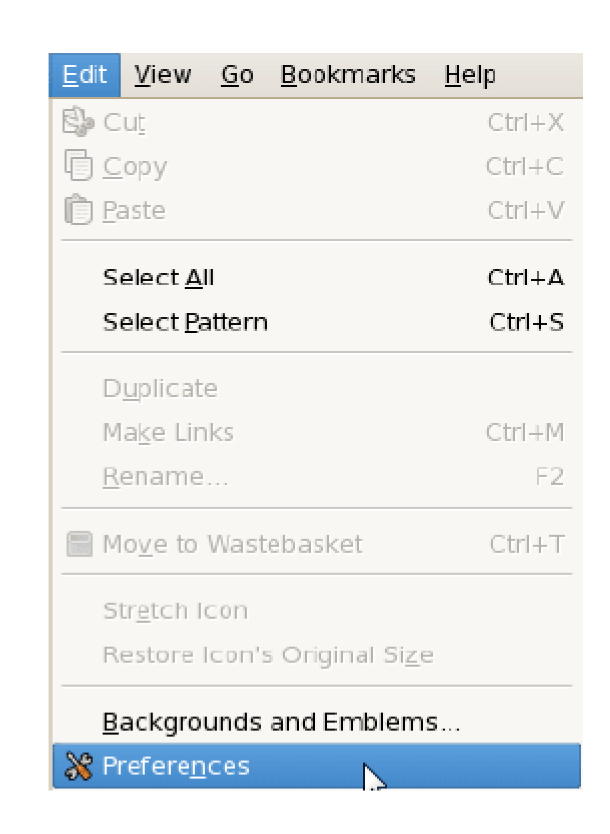


图 2-73　选择 Preferences(首选项)命令

图 2-74　File Management Preferences (文件管理器首选项)对话框

- Behavior(行为)选项卡：可以设置行为、可执行文件、回收站等相关功能，如图 2-75 所示。
- Display(显示)选项卡：可以设置图标标题、日期等功能，如图 2-76 所示。

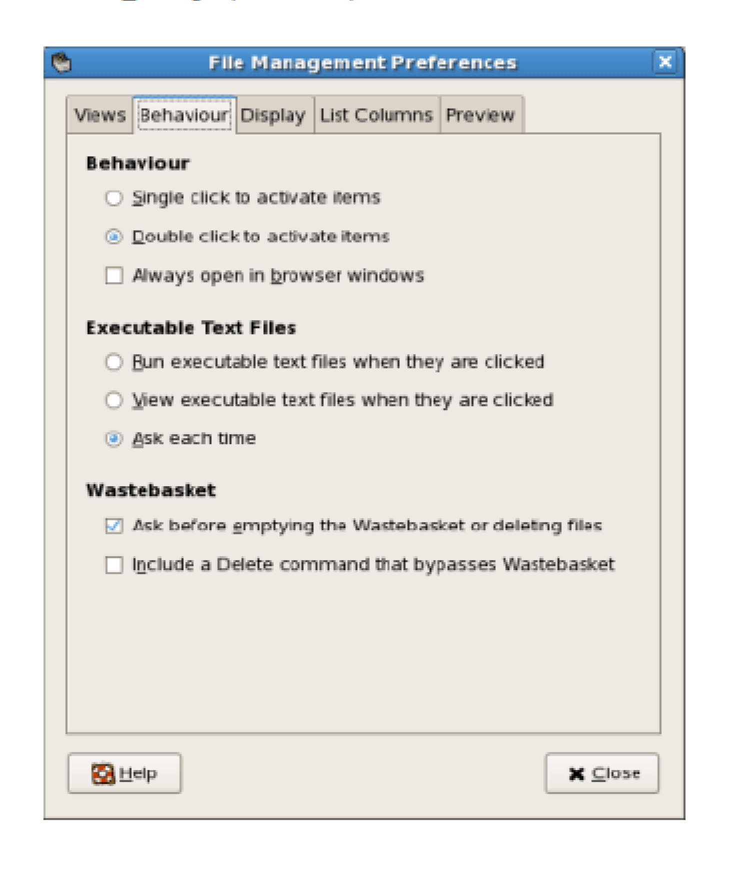


图 2-75　Behaviour(行为)选项卡

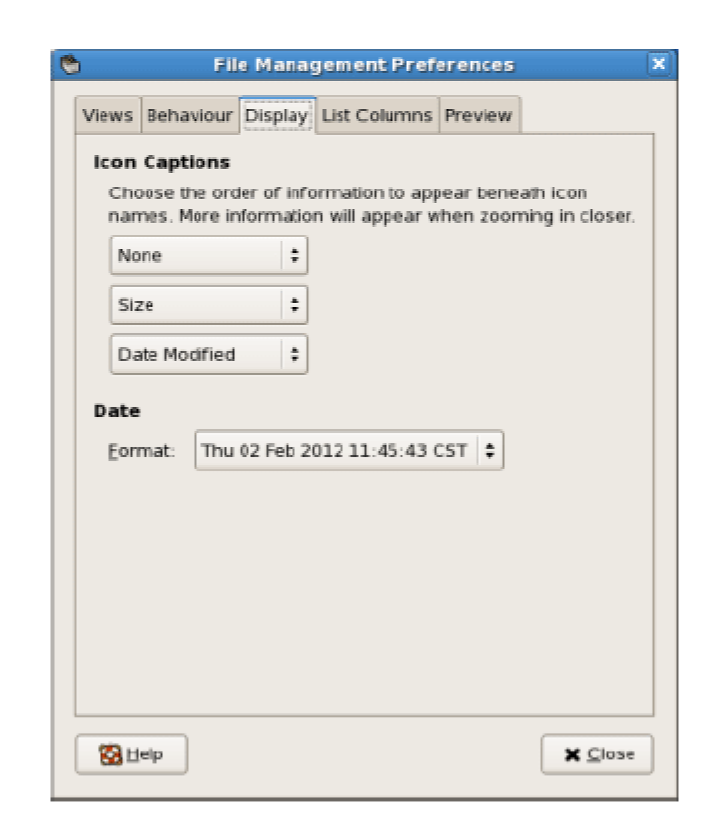


图 2-76　Display(显示)选项卡

- List Columns(列表列)选项卡：可以设置列表的显示项目，如图 2-77 所示。
- Preview(预览)选项卡：可以设置文件的预览效果等功能，如图 2-78 所示。

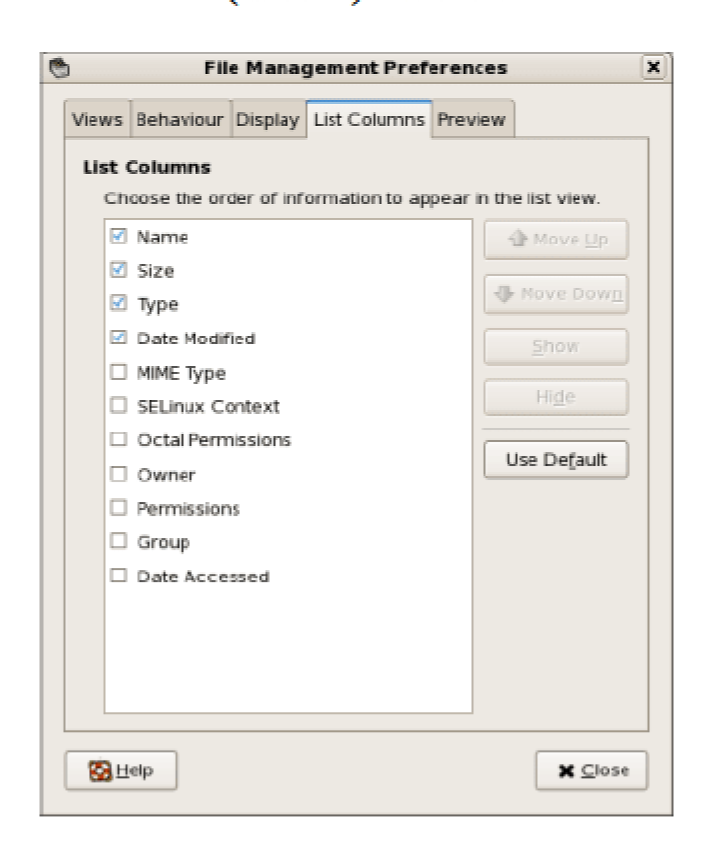


图 2-77　List Columns(列表列)选项卡



图 2-78　Preview(预览)选项卡

修改完成后，只需要单击 Close(关闭)按钮关闭，修改的功能即可随之更改。

5. 运行应用程序

在 GNOME 桌面环境中，运行应用程序主要有以下几种方法。

- 在“面板”中单击 Application 菜单，在菜单中选择需要运行的应用程序。
- 在桌面上双击应用程序的图标。
- 在面板的快速启动区中单击应用程序项目。
- 在虚拟终端窗口中输入命令以启动应用程序。
- 在文件管理器中找到应用程序的可执行文件，双击此文件以运行应用程序。

默认情况下，GNOME 的桌面非常简洁，只有计算机、用户文件夹以及回收站 3 个应用程序的图标，如果用户希望将某个应用程序放在桌面上，便于随时使用，可以通过建立应用程序启动器的方法来完成，具体操作步骤如下。

(1) 在桌面的空白处单击鼠标右键，在弹出的快捷菜单中选择 Create Launcher(创建启动器)命令，如图 2-79 所示。

(2) 弹出 Create Launcher(创建启动器)对话框。在前面的内容中我们已经介绍了如何使用创建启动器对话框在面板中创建一个“关机”快速启动器，下面我们再以创建一个图像浏览器为例，具体的使用方法在此不再介绍，设置完成后的 Create Launcher(创建启动器)对话框如图 2-80 所示。

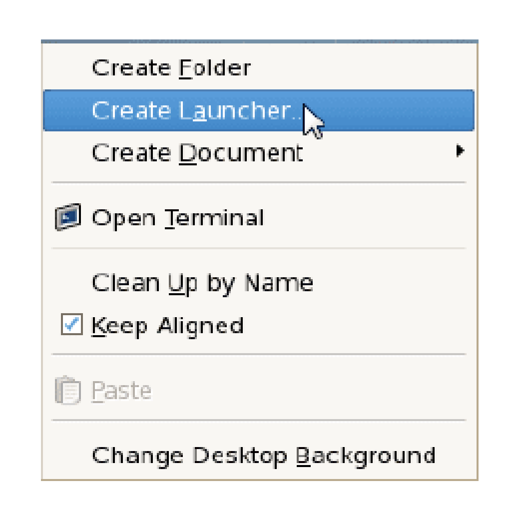


图 2-79 选择 Create Launcher(创建启动器)命令

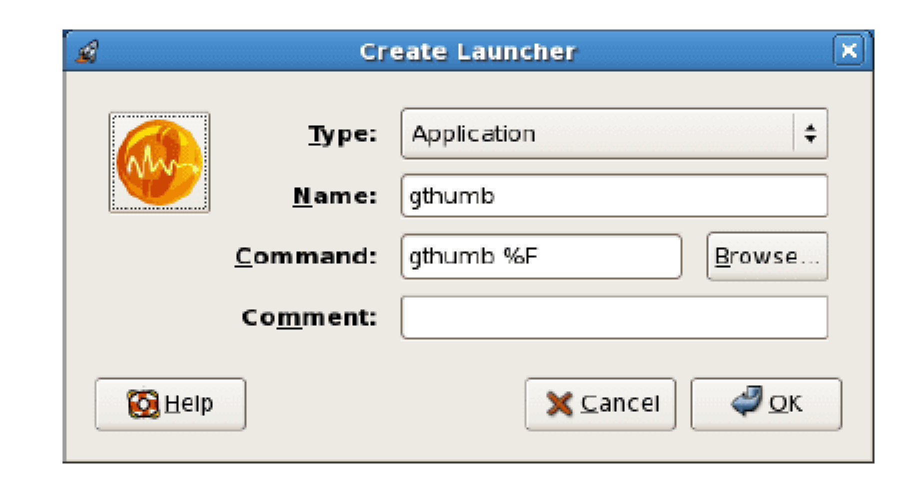


图 2-80 Create Launcher(创建启动器)对话框

(3) 单击 OK 按钮，完成应用程序图标的添加，我们可以在桌面上看到新添加的应用程序图标如图 2-81 所示。

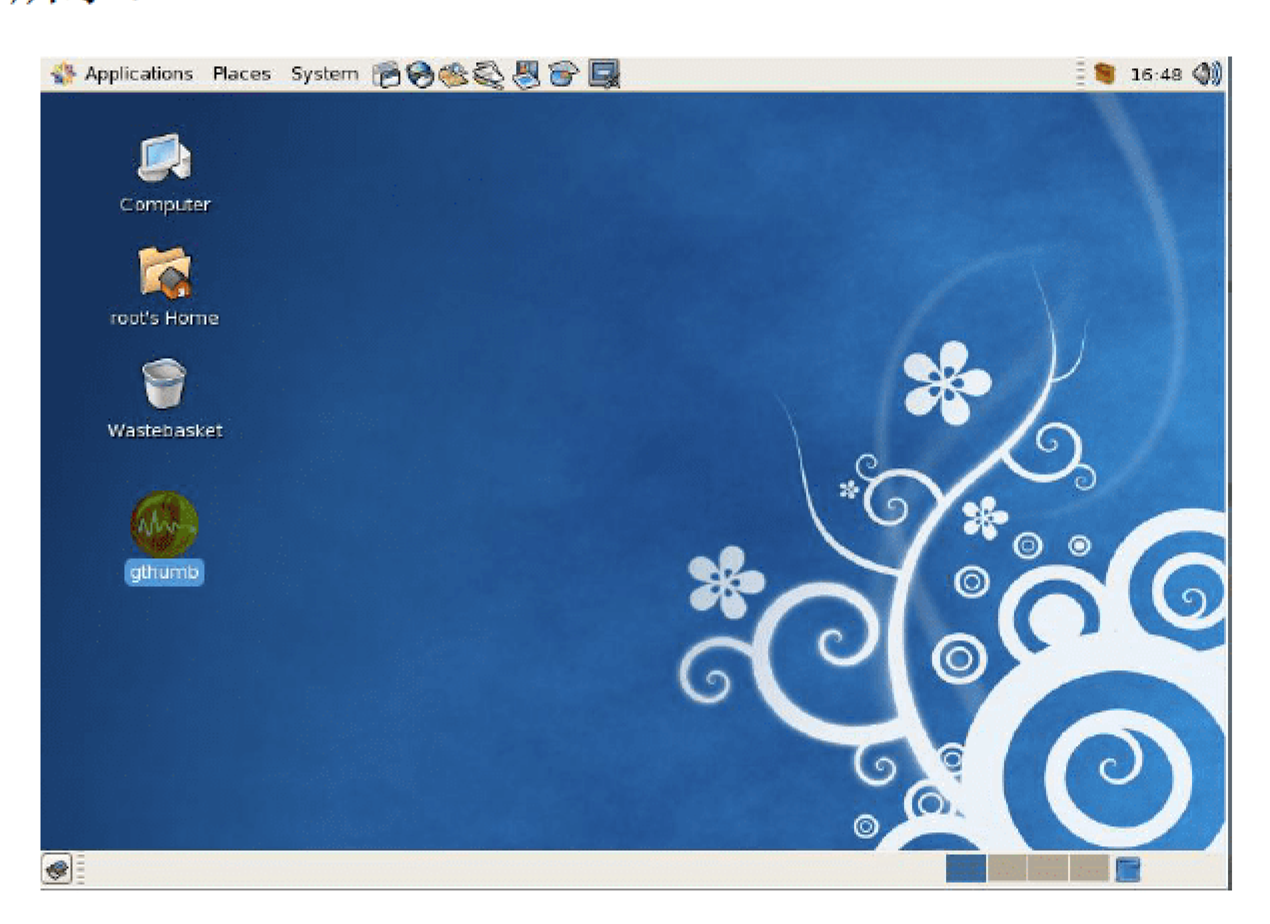


图 2-81 新添加的应用程序图标

另外，在 GNOME 中，大部分的设置工具都集中在面板的 System→Preferences 子菜单中，包括对键盘、鼠标、字体、桌面背景等各种设置功能，如图 2-82 所示。如果用户要对这些项目进行设置，只需要通过命令打开对应的设置窗口进行设置就可以了。

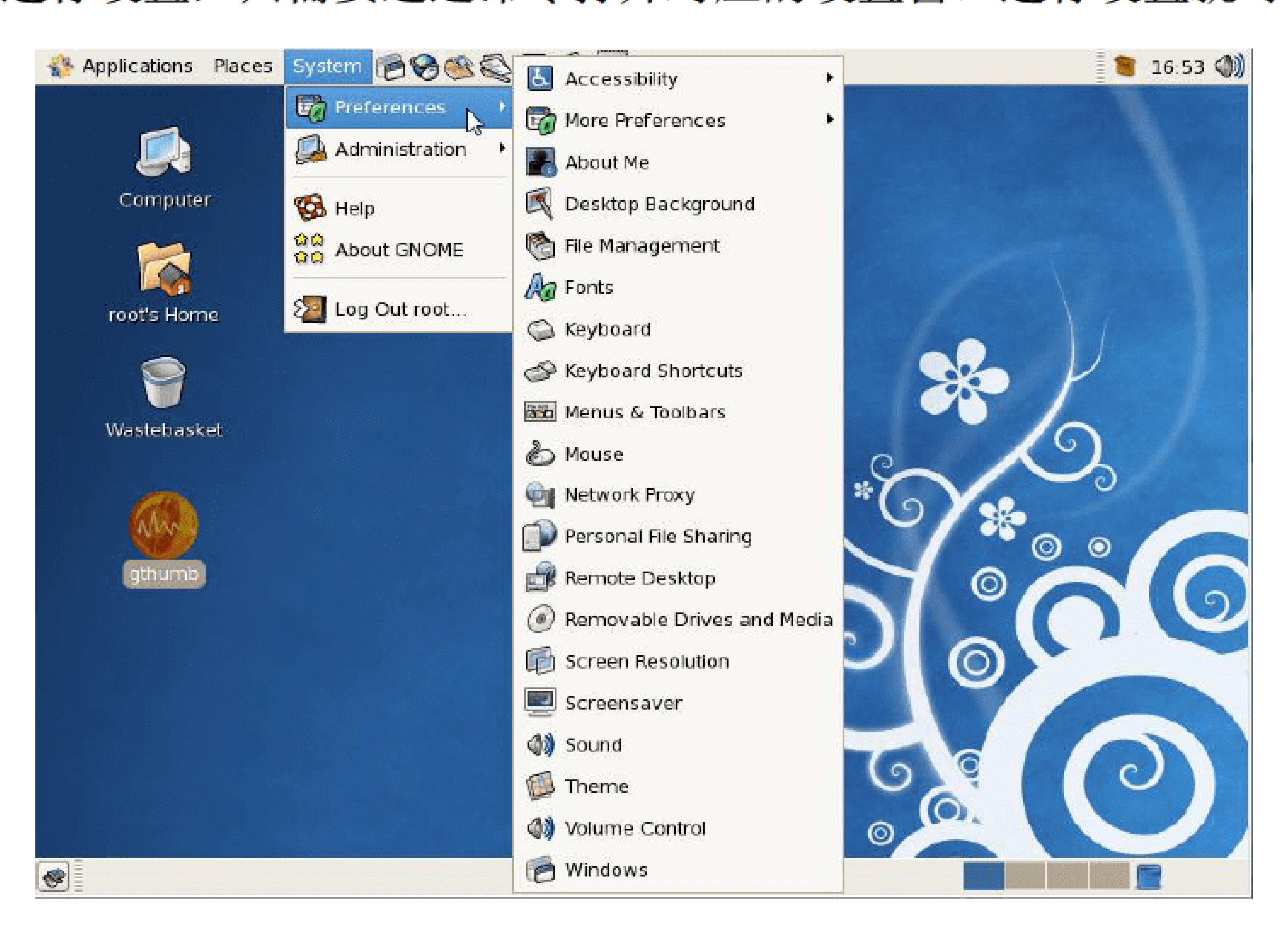


图 2-82　Preferences(首选项)命令

2.5　本 章 小 结

本章中，我们主要介绍了安装 Linux 的基本过程，其中，使用虚拟机来安装 Linux 对学习 Linux 的过程有很大的帮助。另外，我们还着重介绍了 Linux 下的图形界面环境——GNOME 桌面环境的启动方法和基本的使用方法，使读者对 Linux 的图形环境有了基本的了解。

2.6　课 后 习 题

1. 填空题

(1) Linux 常用的安装方法有 5 种：________、__________、________、___________、_________。

(2) 在 Linux 系统中，以______方式访问设备。

2. 选择题

(1) 系统用默认值安装完之后，其安全程度(　　)。

A. 高　　B. 低

C. 视出品公司而定　　D. 仍有安全顾虑，要再进行校调

(2) 下列不是 Linux 系统进程类型的是(　　)。

A. 交互进程　　B. 批处理进程　　C. 守护进程　　D. 就绪进程

(3) 一台 PC 上可以有两个 IDE 接口(将其称为第一 IDE、第二 IDE)，而每个 IDE 接口上可以接两个 IDE 设备(将其称为主盘、从盘)。在 Linux 中，对第二 IDE 的主盘的命名名称为(　　)。

A. /dev/hda　　B. /dev/hdb　　C. /dev/hdc　　D. /dev/hdd

3. 判断题

(1) X Window 系统是 Unix 上的标准图形界面，是一个支持多种应用程序的环境。Linux 用的 X Window 版本通常是 XFree86。 (　　)

(2) 一个硬盘最多能够被分成 2 个主分区。 (　　)

4. 简答题

(1) 简述 Linux 分区与 Windows 分区的不同。

(2) 简述 Linux 的安装过程。

(3) 在 Linux 系统的 GNOME 桌面环境下，如何增加和删除面板？

第 3 章

Linux 常用配置命令

本章将重点介绍 Linux 下的命令及其使用方法。可以说，命令是学习 Linux 必须熟练掌握的基本知识之一。Linux 下的命令大概有 600 多个，但常用的命令其实只有 80 个左右，而这些常用的命令是必须灵活掌握的。虽然 Linux 拥有众多的发行版本，但是其常用命令几乎都是不变的，因此，只要掌握了这些常用的命令，就能够融会贯通 Linux 的使用。

3.1 Linux 下的 shell 介绍

在介绍 Linux 下丰富的命令之前，我们必须首先了解 Linux 下 shell 的基本概念，因为 shell 是 Linux 命令行模式的基本载体。

3.1.1 shell 的基本概念

在计算机科学中，shell 俗称壳(用来区别于核)，是指“提供使用者使用界面”的软件(命令解析器)。它接收用户命令，然后调用相应的应用程序。同时它又是一种程序设计语言。作为命令语言，它交互式解释和执行用户输入的命令或者自动地解释和执行预先设定好的一连串的命令；作为程序设计语言，它定义了各种变量和参数，并提供了许多在高阶语言中才具有的控制结构，包括循环和分支。

各种操作系统都拥有自己的 shell。以 DOS 操作系统为例，它的 shell 就是 command.com 程序。除了它，DOS 操作系统还出现过很多第三方的命令解释程序，例如 4DOS、NDOS 等，这些命令解释程序完全可以取代 command.com 程序。而在 Linux 下，最常见的 shell 是 bash(Bourne Again Shell)。除了 bash，还有 C shell、Korn shell、Bourne shell 和 Tenex C shell 等。每个版本的 shell 功能基本相同，但各有优缺点，现在 Linux 的发行版本一般都以 bash 作为默认的 shell。

实质上，shell 本身是使用 C 语言编写的程序，是系统的用户界面，它提供了用户与内核进行交互操作的一种接口。它接收用户输入的命令并把它送入内核去执行。shell 不仅是一种命令解释程序，还是一种功能强大的解释型程序设计语言，它定义了各种选项和变量，几乎支持高级编程语言的所有程序结构，例如变量、函数、表达式和循环等。利用 shell 可以编写程序，生成功能强大的控制命令和脚本。

提示： Linux 下的 shell 编程又是一个全新的领域，本书中不做详细介绍，有兴趣的读者可以参考其他相关书籍。

在大多数 shell 中还定义了一些内置的命令。这主要是为了加快命令的运行，并且更有效的定制 shell 程序。我们将 shell 自身解释执行的命令称为内置命令，例如管理员会经常用到的 cd、pwd、exit 和 echo 等都属于 bash 的内置命令。当用户登录系统后，shell 及其内置命令就被系统加载到内存中，并且一直运行、直到用退出操作系统为止。除了内置命令，在 Linux 系统中还有很多可执行文件，这些可执行文件类似于 Windows 下的 exe 文件，这些可执行文件同样可以作为 shell 命令来执行，例如 ls 命令就是一个可执行文件，存放在/bin/ls 中。其实很多 Linux 的命令都不是 shell 的内置命令，这些命令与内置命令不同，只有当它们被调用的时候才会由系统载入内存中。表 3-1 给出了 Linux 系统中的可执行文件分类的说明。

当用户以系统的默认级别(级别 3)登录 Linux，将直接进入字符界面，即 shell 的命令提示符界面。提示符格式如下：

```
[root@CentOS ~]#
```

其中，“root”代表登录用户名，“centOS”为当前登录的服务器名，“#”说明此用户为超级用户。

表 3-1　Linux 系统中的可执行文件分类

类　别	说　明
内置命令	构造在 shell 内部
Linux 命令	存放在/bin、/sbin 目录中的命令
实用程序	存放在/usr/bin、/usr/sbin、/usr/share、/usr/local/bin 等目录下的实用程序或者工具
用户程序	用户将程序经过编译后生成可执行文件，也可以作为 shell 命令执行
Shell 脚本	由 shell 语言编写的批处理文件

shell 执行命令解释的具体过程为：用户在命令行中输入命令并且提交后，shell 程序首先检测其是否为内置命令，如果是，就通过 shell 内部的解释器将命令解释为系统调用，然后提交给内核执行；如果不是 shell 内置命令，那么 shell 会按照用户给出的路径或者根据系统的环境变量在硬盘中寻找对应的命令，然后将其调入内存，将其解释为系统调用，提交给内核执行。如图 3-1 给出了 shell 对命令的解释过程。

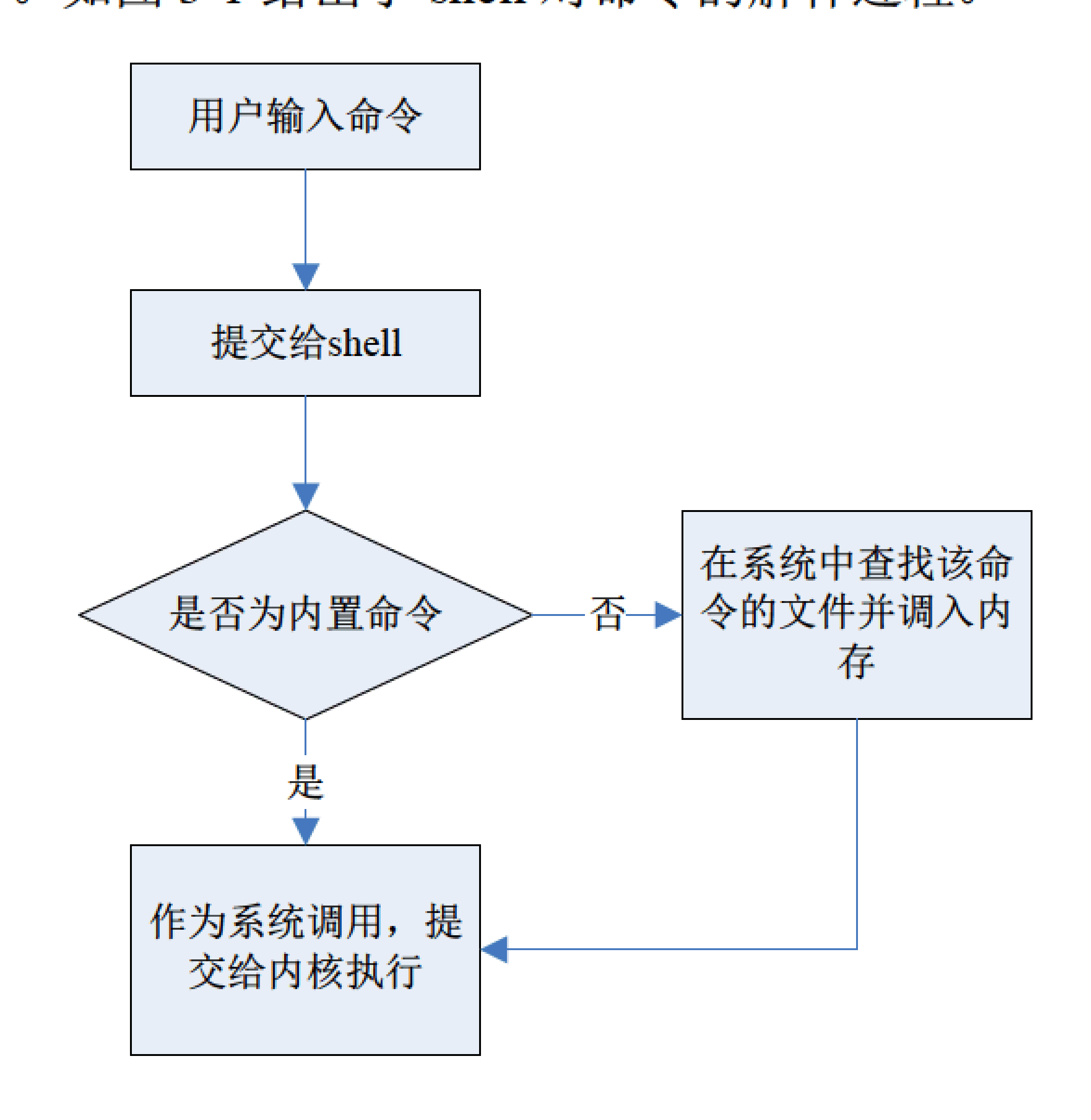


图 3-1　shell 对命令的解释过程

3.1.2　shell 命令语法说明

Linux 下的各种 shell 的主要区别在于命令行的语法。但对于一些普通的命令来说，各个版本的 shell 语法基本相同。只有在编写 shell 脚本或者使用 shell 的高级特性时，才会遇到差异。

shell 的语法分析是指 shell 对命令的扫描处理过程，也就是把命令或者用户输入的内容分解称为要处理的各个部分。在 Linux 系统下，shell 语法分析包括很多的内容，例如重定向、文件名扩展和管道等。

下面我们将以 bash 为例介绍 shell 命令的语法分析。

1. shell 的命令格式

shell 是遵循移动的语法格式将用户输入的命令进行分析解释并传送给系统内核的。shell 的一般格式为：

```
command [options] [arguments]
```

其中：

- command 是命令的名称。
- options 是命令的选项。
- arguments 是命令的参数。

我们将具有以上格式特征的字符串称为命令行。命令行是用户与 shell 之间交互的一种基本单位。在命令行中选项(Options)是包含一个或者多个字母的代码，主要用于改变命令的执行方式，一般情况下，选项前面会带有“-”符号，用来与参数进行区分。例如以下的命令：

```
[root@CentOS ~]#ls -a
```

ls 命令附带“-a”选项后，表示列出当前目录下的所有文件(包含隐藏文件)。否则，ls 命令只会列出当前目录下的文件名和目录，而不显示隐藏文件。

很多命令都包含多个选项，这些选项既可以单独列出，也可以在“-”后将所需要的选项依次都列出来，例如以下命令的两种书写方式是等价的。

```
[root@CentOS ~]#ls -a -l
[root@CentOS ~]#ls -al
```

参数(Arguments)是指紧跟在选项后的一个或者多个字符，这些字符指定了命令的操作对象，文件或者目录等。例如，要显示/etc 目录下的所有文件的信息，可以使用以下的命令：

```
[root@CentOS ~]#ls -al /etc
```

有些命令的参数是可选的(例如 ls 命令)，而有些命令是必须附带若干参数的，否则，shell 会给出错误提示信息。例如，移动文件或者更名的命令——mv 命令就必须附带两个参数，用来指明命令操作的源文件和目标文件。

在一个命令行中，还可以一次执行多个命令，只需要将这些命令用分号分开，shell 就会在解释命令的时候依次执行。例如，下面的命令行同时执行了 ls 命令和 cp 命令。

```
[root@CentOS ~]#ls -al;cp mufile.txt myfile1.txt
```

当然，我们也可以将一条命令在多个命令行中输入进行执行，用“\”就可以将一条命令持续到下一行继续输入，如下面的例子：

```
[root@CentOS ~]#cp -I \
>myfile.txt\
>myfile1.txt
```

另外，bash 还可以自动补齐命令行，即在输入命令的时候不必把命令全部输完，shell 能够智能判断用户所需要输入的命令。当用户输入某个命令的一部分后，按 Tab 键，shell

就可以根据系统环境变量信息提示出与用户输入命令相似的所有命令和文件，例如：

```
[root@CentOS ~]#if  <按 Tab 键>
if   ifcfg   ifconfig   ifdown   ifenslave   ifnames   ifrename   ifup
[root@CentOS ~]#if
```

从提示信息中我们可以看出，用户输入“if”后按 Tab 键，即可显示出所有以“if”为起始的所有命令和可执行文件，而如果管理员希望输入的是 ifconfig，那么只需要输入“ifco”再按 Tab 键，shell 就会自动补全命令。

如果管理员需要进入或者输入一个很深的目录中的内容，并且目录名又比较长，此时使用 bash 的自动补全功能可以节省大量的输入时间。

2. shell 通配符的使用

通配符是一些特殊的字符，其目的主要是为了方便用户对文件或者目录的描述，用户可以在命令行的参数中使用这些字符进行文件名或者路径的匹配，shell 将会把命令行中指定的符合要求的所有文件名或者路径名作为命令的参数来执行命令。例如用户只需要以.sh 为后缀时。就可以使用通配符来实现。

各个版本的 shell 通配符规则基本相同，下面我们还是以 bash 为例介绍通配符的使用方法。在 bash 中，常用的通配符有“*”、“？”、“[]”。

1) “*”通配符

“*”通配符用来匹配任意一个或者多个字符。例如：

```
[root@CentOS ~]#ls *.sh
```

此命令将列出当前目录中所有以“.sh”结尾的所有文件。又如：

```
[root@CentOS ~]#ls -al /usr/*/*.sh
```

此命令是列出/usr 目录的所有子目录中以“.sh”结尾的文件。

2) “?”通配符

“?”通配符匹配任意一个字符。使用方法如下面的两个例子。

```
[root@CentOS ~]#ls ab?.txt
[root@CentOS ~]#ls ab??.txt
```

其中，第一条命令是列出当前目录下所有以 ab 开头，第三个字母为任意符号，然后以“.txt”为后缀的文件，而第二条命令是列出当前目录下所有以 ab 开头，第三、四个字母为任意符号，然后以“.txt”为后缀的文件。

3) “[]”通配符

“[]”通配符匹配包含在方括号内的单个字符。以下面的命令为例

```
[root@CentOS ~]#ls /dev/sda[12345]
/dev/sda1   /dev/sda2   /dev/sda3
```

从上面的例子我们可以看出，命令执行后，系统列出了所有在/dev 目录下以盛大开头，第四个字符是 1、2、3、4 和 5 中任意一个字符的文件。

另外，当“[]”通配符中列举的数字为连续数字的时候，也可以以范围的方式表示，如下面的例子所示，两种表现方法完全等效。

```
[root@CentOS ~]#ls /dev/sda[1-5]
```

4) 通配符的组合使用

为了复杂匹配的需要，通配符还可以组合使用。例如下面的两个例子：

```
[root@CentOS ~]#ls [0-7]??ab.conf
[root@CentOS ~]#ls [abcd]*.sh
```

读者可以根据前面讲述的各种匹配符的含义和使用方法，试着给出这两条命令的具体含义。

3. shell 重定向

所谓的重定向，就是不使用系统默认的标准输入输出，而是重新指定新的输入输出。Linux 系统具有标准输入、标准输出和标准错误输出。用户的 shell 将键盘作为默认的标准输入，默认的标准输出和标准错误输出为屏幕。也就是用户从键盘输入命令，然后将结果和错误信息输出到屏幕。所以，重定向也可以分为输入重定向、输出重定向和错误输出重定向。要实现重定向就需要了解重定向操作符，shell 就是根据重定向操作符来决定重定向操作的。

1) 输入重定向

输入重定向用于改变命令的输入源，利用输入重定向，就可以将一个文件的内容作为命令的输入，而不从键盘输入。用于输入重定向的操作符有“<”和“<<”。如下面的例子所示：

```
[root@CentOS ~]#wc </etc/inittab
53  229 1666
```

wc 命令用来统计输入给它的文件/etc/inittab 的行数、单词数和字符数，然后再屏幕中输出。

还有一种输入重定向符号——“<<”，这种重定向告诉 shell，当前命令的标准输入为来自命令行中一对自定义分隔符之间的内容。如下面的例子所示：

```
[root@CentOS ~]#wc <<aaa
> # Default runlevel. The runlevels used by RHS are:
> #   0 - halt (Do NOT set initdefault to this)
> #   1 - Single user mode
> #   2 - Multiuser, without NFS (The same as 3, if you do not have
networking)
> #   3 - Full multiuser mode
> #   4 - unused
> #   5 - X11
> #   6 - reboot (Do NOT set initdefault to this)
> aaa
  8  65 303
```

上面的命令将一对分隔符 aaa 之间的内容作为 wc 命令的输入。分隔符可以是任意字符。shell 将在第一个分隔号后开始读取内容，直到出现另一个分隔号读取结束，然后将内容送给 wc 命令处理。

提示： 分隔符最好定义为不常见的符号或者符号组合，特别注意在分隔符之间的内容一定不能出现与分隔符相同的组合，否则输入将提前结束。

2) 输出重定向

输出重定向是将命令的输出结果不在屏幕输出，而是输出到一个指定文件中。在 Linux 下输出重定向用得很多，例如，某个命令的输出很长，一个屏幕无法显示完毕，我们可以将命令的输出指定到一个文件，然后用 more 命令查看这个文件，从而得到命令输出的完整信息。

用于输出重定向的操作符有“>”和“>>”。下面给出了使用“>”操作符的例子。

```
[root@CentOS ~]#ps -ef >ps.txt
```

将 ps －ef 输出的系统运行进程信息全部输入到了 ps.txt 文件，而不输出到屏幕，可以用 more 命令查看 ps.txt 文件中系统运行的进程信息。如下面的命令方式：

```
[root@CentOS ~]#more file1 file2 file3 >file
```

上面的命令是将 file1、file2 和 file3 的内容全部输出到 file 文件中，类似于合并文件内容的功能。

如果在“>”后面指定的文件不存在的话，shell 就会自动创建；如果文件存在的话，那么这个文件原有的内容将被覆盖；如果不想覆盖存在的文件，可以使用“>>”操作符。例如：

```
[root@CentOS ~]#ls -al /etc/* >>/root/install.log
```

此命令将/etc 目录及其子目录下的所有文件信息追加到/root/install.log 文件的结尾。/root/install.log 文件原来的内容仍然存在。

3) 错误重定向

错误重定向和标准输出重定向一样，可以使用操作符“2>”和“2>>”实现对错误输出的重定向。如下面的例子：

```
[root@CentOS ~]#tar zxvf text.tar.gz 2> error.txt
```

tar 是打包命令，可以在屏幕上看到 tar 的解压过程。如果“text.tar.gz”是个损坏的压缩包文件，把错误信息会输出到 error.txt 文件中。

4．shell 管道

管道可以把很多命令连接起来，把一个命令的输出当作下一个命令的输入，而不经过任何中间文件。例如可以将第 1 个命令的输入当作第 2 个命令的输出，而将第 2 个命令的输出当作第 3 个命令的输入。

通过管道符“|”可以建立一个管道连接，例如下面的命令：

```
[root@CentOS ~]# ls -al /etc/* | more
```

表示将/etc 目录以及子目录下的所有文件分屏显示。

```
[root@CentOS ~]#ps -ef|grep httpd|wc -l
```

这个命令是查看系统中正在运行的 httpd 进程，并计算 httpd 的进程数。

5．shell 引用

在 bash 中有很多特殊字符，这些字符本身就具有特殊含义。如果在 shell 的参数中使

用它们，就会出现问题。Linux 中使用了“引用”技术来忽略这些字符的特殊含义，引用就是指通知 shell 将这些特殊字符当作普通字符处理。shell 中用于引用的字符有转义字符“\”、单引号“'”、双引号“"”。

1) 转义字符“\”

如果将“\”放到特殊字符前面，shell 就忽略这些特殊字符的原有含义，当作普通字符来处理，例如：

```
[root@CentOS ~]#ls
abc?*  C:\backup
[root@CentOS ~]#mv abc\?\*  abc
[root@CentOS ~]#mv C\:\\backup backup
```

上面是将 abc?*重命名为 abc，将 C:\backup 重命名为 backup。因为文件名中含有特殊字符，所有都使用了转义字符“\”。

2) 单引号“''”

将字符串放到一对单引号之间，那么字符串中所有字符的特殊含义将被忽略，例如：

```
[root@WEBServer ~]#mv C\:\\backup backup
[root@WEBServer ~]#mv 'C:\backup'  backup
```

上面两条命令的功能完全等效。

3) 双引号“""”

双引号的引用与单引号基本相同，包含在双引号内的大部分特殊字符可以当作普通字符处理，但是仍有一些特殊字符即使使用双引号括起来，也仍然保留自己的特殊含义，比如“$”、“\”和“''”。请看下面的例子：

```
[root@CentOS ~]#str="The \$SHELL Current shell is $SHELL"
[root@CentOS ~]#str1="\$$SHELL"
[root@CentOS ~]#echo $str
The $SHELL Current shell is /bin/bash
[root@CentOS ~]#echo $str1
$/bin/bash
```

从上面输出可以看出，“$”和“\”在双引号内仍然保留了特殊含义。又如：

```
[root@CentOS ~]# str="This hostname is `hostname`"
[root@CentOS ~]# echo $str
This hostname is WEBServer
```

上面的输出中，字符“`”在双引号中也保留了自己特殊含义。

3.2 Linux 常用命令及使用

Linux 拥有众多的命令，下面我们将对其分类进行介绍。

3.2.1 系统管理类

1. ls 命令

ls 命令用来显示指定工作目录下的内容，列出工作目录所包含的文件及其子目录。另

外，Linux 还提供了 dir 命令(与 DOS 操作系统的命令类似)可以用来替代 ls 命令，ls 命令的语法格式如下：

```
ls [选项] [路径或者文件名]
```

表 3-2 给出了 ls 可以使用的选项及其具体含义。

表 3-2　ls 命令选项及其含义

选　项	含　义
-a	显示指定目录下的所有文件和子目录，包含隐藏文件(在 Linux，所有以“.”开头的文件或者目录都被认为是隐藏文件)
-d	只显示目录列表，不显示文件
-l	除文件名称以外，同时列出文件或者子目录的权限、使用者和大小等信息
-s	在每个文件名后输出该文件的大小
-k	以 k 字节的形式表示文件的大小
-u	以文件上次访问的时间排序
-t	以时间排序
-o	显示除组信息外的详细信息
-x	按列输出、横向排列
-r	对目录反向排序
-q	用?代替不可以输出的字符
-m	横向输出文件名，并以“,”作为分隔符
-S	以文件大小排序
-R	列出素有子目录下的文件
-pf	在每个文件后附加一个字母以说明该文件的类型。“*”表示可执行的普通文件。“/”表示目录，“@”表示符号链接，“|”表示 FIFOs，“=”表示套接字(sockets)
-C	按列输出，纵向排列
-Q	把输出的文件名用双引号括起来

例如，要列出/usr 目录下的文件及其子目录的详细信息，可以使用以下命令：

```
[root@CentOS /]# ls -l /usr
total 256
drwxr-xr-x   2 root root 69632 Feb  1 18:35 bin
drwxr-xr-x   2 root root  4096 Jan 27  2010 etc
drwxr-xr-x  11 root root  4096 Jan 18 23:21 include
drwxr-xr-x   6 root root  4096 Jan 18 23:12 kerberos
```

又如，如果要显示/usr/local 下的所有文件及其子目录的详细信息，包括文件类型的标记，那么可以使用一下命令：

```
[root@CentOS /]# ls -alF /usr/local
total 88
drwxr-xr-x 11 root root 4096 Jan 18 23:09 ./
drwxr-xr-x 14 root root 4096 Jan 18 23:11 ../
drwxr-xr-x  2 root root 4096 Jan 27  2010 bin/
drwxr-xr-x  2 root root 4096 Jan 27  2010 etc/
```

```
drwxr-xr-x  2 root root 4096 Jan 27  2010 games/
drwxr-xr-x  2 root root 4096 Jan 27  2010 include/
drwxr-xr-x  2 root root 4096 Jan 27  2010 lib/
drwxr-xr-x  2 root root 4096 Jan 27  2010 libexec/
drwxr-xr-x  2 root root 4096 Jan 27  2010 sbin/
drwxr-xr-x  4 root root 4096 Jan 18 23:09 share/
drwxr-xr-x  2 root root 4096 Jan 27  2010 src/
```

2. pwd 命令

pwd 命令用来显示当前的工作目录，用户输入 pwd 命令即可知道目前所在工作目录的绝对路径名，如下面的例子：

```
[root@CentOS bin]# pwd
/usr/local/bin
```

3. cd 命令

cd 命令可以改变当前的工作目录。具体语法如下：

```
cd [目录名]
```

cd 命令的选项及其说明如表 3-3 所示。

表 3-3　cd 命令选项及其说明

选　项	含　义
cd [目录名]	切换到目录名指定的目录下(注意 Linux 目录名的对大小写敏感)
cd 或者 cd ~	返回当前用户的默认工作目录
cd~[用户名]	切换到指定用户的工作目录下
cd ..或者 cd ../	返回到上一级目录
cd /	返回到根目录

4. date 命令

date 命令用来显示或者修改系统的时间和日期。普通用户智能使用 date 来查看系统时间。只有超级用户才有权限使用 date 命令来修改日期。date 命令的语法如下所示：

```
date [选项] 显示时间格式(以“+”开始)
```

date 命令的选项以及含义如表 3-4 所示。

表 3-4　date 命令选项及其说明

选　项	含　义
-s 或者--set	设置系统时间
-d 或者--date	按照显示时间格式来显示系统时间

另外，date 命令可以设置的显示时间格式非常丰富，可使用的格式及其说明如表 3-5 所示。

表 3-5　date 命令显示时间格式

格　式	含　义
%H	显示小时，显示格式为 00~23
%I	显示小时，显示格式为 01~12
%k	显示小时，显示格式为 0~23
%l	显示小时，显示格式为 1~12
%M	显示分钟，显示格式为 00~59
%S	显示秒钟，显示格式为 00~59
%p	显示 AM(上午)或者 PM(下午)
%r	显示时间格式设为 hh:mm:ss，显示 AM 或者 PM
%T	显示时间格式设为 hh:mm:ss
%x	显示年份和日期，格式设为 mm/dd/yyyy
%X	显示时间，格式相当于%H:%M:%S 的组合
%a	显示星期
%b 或者%B	显示月份，%b 显示月份的英文简称，%B 为英文全称
%m	显示月份，设定格式为 01~12
%c	显示日期和时间格式为%a
%Z	显示时区
%d	显示为一个月的第几天
%D	显示年份和月份，格式为 mm/dd/yy，yy 表示年份的最后两位
%Y 或者%y	显示年份，%Y 为显示完成的年份，%y 为显示年份的最后两位
%c	显示日期和时间，格式为“%a %d %b %Y %r %Z”的组合

例如，我们如果要在显示时间的过程中写入一些说明性的文字，可用如下的命令格式：

```
[root@CentOS /]# date '+today is %x %a,the time is %X.'
today is 08/02/12 Wed,the time is 11:10:17.
```

如果要修改系统时间，首先要确认登录用户具有超级权限，然后使用“date –s”来设定时间。具体的设定时间的表示方式如下面的例子所示：

```
[root@CentOS /]#date -s 20110505     #设定系统日期
[root@CentOS /]#date -s 15:00         #设定系统时间
[root@CentOS /]#date -s "20110505 15:00"   #设定系统时间和日期
```

另外，我们也可以使用“date –d”来设定描述的日期。如下面的例子所示：

```
[root@CentOS /]# date '+%Y-%m-%d'              #显示当前日期
2012-02-08
[root@CentOS /]# date -d "5 days ago" +%Y-%m-%d   #显示 5 天前的日期
2012-02-03
```

5. passwd 命令

passwd 命令用来修改用户的密码。普通用户只能修改自己的用户密码，超级用户则

可以修改自己和普通用户的密码。语法格式为：

```
passwd [用户名]
```

修改密码的范例如下所示：

```
[root@CentOS /]#passwd      #修改自己的密码
[root@CentOS /]#passwd root    #root 用户修改自己的密码
[root@CentOS /]#passwd [用户名]    #超级用户修改其他用户的密码
```

输入命令后，系统会要求用户输入两次新的密码，输入成功后，新的密码就立即生效。

6. su 命令

su 命令用来改变用户身份。命令格式为：

```
su [选项] [用户名]
```

su 命令的可用选项及其具体含义如表 3-6 所示。

表 3-6 su 命令选项及其含义

选　项	含　义
-	加载相应用户的环境变量
-l	使目前的 shell 成为改变用户后的默认 shell
-m	改变用户身份，但是不改变环境变量

例如，普通用户如果想成为超级用户，可以使用以下两个命令中的一个：

```
[root@CentOS /]#su -     #提升权限，并且加载 root 环境变量
[root@CentOS /]#su       #只提升权限，不加载环境变量
```

然后根据系统提示输入超级用户的密码即可成为超级用户。

注意： “su”与“su –”命令的区别在于，“su”命令没有加载 root 环境变量，因此某些命令因为找不到路径而可能无法执行。

7. clear 命令

clear 命令用来清除用户的屏幕信息。使用格式为：

```
clear
```

8. man 命令

Man 命令用来显示指定命令的帮助信息。其格式为：

```
man [命令名称]
```

例如，想要获得关于 clear 命令的帮助信息，输入的命令及其显示的信息如下：

```
[root@CentOS /]#man clear
NAME
      clear - clear the terminal screen
```

```
SYNOPSIS
       clear

DESCRIPTION
       clear clears your screen if this is possible.  It looks in the
environ-
       ment for the terminal type and then in the terminfo database to
figure
       out how to clear the screen.

       clear ignores any command-line parameters that may be present.

SEE ALSO
       tput(1), terminfo(5)

       This describes ncurses version 5.5 (patch 20060715).

 (END)
```

9. who 命令

who 命令用来显示当前登录到系统中的用户及其信息，使用格式为：

```
who [选项] [file]
```

who 命令的选项及其含义如表 3-7 所示。

表 3-7　who 命令选项及其含义

选　项	含　义
-a	列出所有信息，即所有选项
-b	列出系统最新启动的时间日期
-l	列出素有可登录的终端信息
-m	仅列出关于当前终端的信息。相当于“who am I”命令
-q	列出在本地系统上的用户和用户的清单
-r	显示当前系统的运行级别
-s	仅显示名称、线路和时间字段的信息
-u	显示当前每个用户的用户名、登录终端、登录时间、线路活动和进程标识
-T 或者-w	显示 tty 终端的信息，其中，“+”标识对任何人可写，“-”标识仅对 root 用户和所有者可写，“?”标识遇到线路故障

who 命令的一般输出格式为：

```
用户名 [状态] 线路 时间 [活动] [进程标识](主机名)
```

各个字段的具体含义为：

- 用户名：用户的登录名。
- 状态：表明线路对用户是否都是可写的。
- 线路：可表示的内容为 tty、pts/1、pts/2 等，这些线路的标识可以在/dev 目录中找到。

- 时间：用户登录系统的时间。
- 活动：某个用户在自己的线路上最后一次活动发生以来到现在的时间。如果此项显示的为“.”，说明一分钟内有线路活动；如果线路保持静止已经超过 24 小时或者自系统启动以来还没有被使用过，此项将显示为“old”。
- 进程标识：用户登录 shell 的进程 id。
- 主机名：登录到 Linux 系统的客户端计算机的名称。

默认情况下，who 命令是通过读取 Linux 系统中的/var/run/utmp 文件来获取需要的信息的。但是如果管理员对 Linux 系统做过相应的修改，希望 who 读取其他的文件，那么可以在[file]选项中指定需要读取的文件。

例如，我们要查看系统的运行级别是什么，可以使用以下的命令：

```
[root@CentOS ~]# who -r
         run-level 3  2012-02-08 10:31                   last=S
```

如果要显示系统最新的启动日期，以及当前每个用户的登录详情和终端状态，可以使用以下的命令：

```
[root@CentOS ~]# who -buT
           system boot  2012-02-08 10:31
root     + tty1         2012-02-08 10:32 02:05        4259
root     + pts/0        2012-02-08 10:35   .         4648 (:0.0)
root     + pts/1        2012-02-08 10:51   .         4753 (210.31.197.88)
```

10. w 命令

W 命令用来显示登录到系统的用户信息，其使用格式为：

```
w [选项] [用户]
```

w 命令可使用的选项及其具体含义如表 3-8 所示。

表 3-8　w 命令选项及其含义

选　项	含　义
-h	不显示输出信息的标题
-l	用长格式输出
-s	用短格式输出，不显示用户登录时间，jCPU 和 PCPU 时间
-V	显示版本信息

如果使用[用户]选项，则说明只列出该用户的信息。

例如，使用 w 命令显示当前用户登录信息的格式为：

```
[root@CentOS ~]# w
 12:50:52 up  2:19,  3 users,  load average: 0.06, 0.01, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
root     tty1     -                 10:32    2:16m  0.71s  0.00s /bin/sh
/usr/bi
root     pts/0     :0.0              10:35    1.00s   0.10s  0.01s w
root     pts/1    210.31.197.88      10:35    1.00s   0.10s  0.01s w
```

显示内容的具体含义为：

- 第一行给出了当前的系统时间，系统自启动到现在已经运行的时间。登录到系统中的用户数和系统的平局负载。平局负载是以 1 分钟、5 分钟、15 分钟内系统的平局负载情况来显示的。
- USER：登录到系统的用户名。
- TTY：用户使用的 TTY 名称。
- FROM：表示用户的登录地点，一般显示为远程登录主机的 IP 地址或者主机名。
- LOGIN@：用户登录的日期和时间。
- IDLE：表示某个程序上次从终端开始执行到现在所持续的时间。
- JCPU：表示该终端所有进程及子进程所使用的系统总时间。
- PCPU：当前活动的进程使用的系统时间。
- WHAT：当前用户执行进程的名称和选项。

11. uname 命令

uname 命令用来显示操作系统的相关信息。具体的使用格式为：

```
uname [选项]
```

uname 命令可使用的选项及其说明如表 3-9 所示。

表 3-9　uname 命令选项及其含义

选　项	含　义
-a	显示操作系统的全部信息
-m	显示系统 CPU 类型
-n	显示操作系统的主机名
-s	显示操作系统类型
-r	显示操作系统内核版本

例如，要查看操作系统的全部信息，使用“uname -a”及其显示信息为：

```
[root@CentOS ~]# uname -a
Linux CentOS 2.6.18-194.el5 #1 SMP Fri Apr 2 14:58:35 EDT 2010 i686 i686
i386 GNU/Linux
```

从上面的信息我们可以看出，操作系统的主机名为 CentOS，内核版本为 2.6.18-194.el5，服务器使用的是 i386 架构的 CPU。

12. uptime 命令

uptime 命令用来输出系统任务队列的信息，输出的信息包括：当前系统时间和系统开机运行的时间，目前有多少用户在线和系统平局负载等。显示的具体信息如下面的例子：

```
[root@CentOS ~]# uptime
 13:21:55 up  2:50,  2 users,  load average: 0.00, 0.02, 0.00
```

13. last 命令

last 命令可以列出目前或者曾经登录系统的用户相关信息。last 命令实际上是读取文

件/var/log/wtmp 中的信息，并把该文件中记录的登入记录全部显示出来。其使用的格式为：

```
last [选项] [-n 显示列数]
```

last 命令的选项及其具体含义如表 3-10 所示。

表 3-10 last 命令选项及其含义

选 项	含 义
-a	将从何处登录系统的主机名或者 IP 地址显示在最后一行
-R	不显示登入系统的主机名或者 IP 地址
-x	显示系统关机、重启以及执行等级的改变等信息
“-n 行数”或者“-行数”	设置列出名单的显示行数
-d	将显示的 IP 地址转换为主机名

例如，要显示最近的 5 条系统的登录信息，并且将 IP 地址显示在最后面的命令及其显示效果如下面的命令所示：

```
[root@CentOS ~]# last -a5
root     pts/0        Wed Feb  8 10:35   still logged in    :0.0
root     tty1         Wed Feb  8 10:32   still logged in
reboot   system boot  Wed Feb  8 10:31          (03:06)     2.6.18-194.el5
root     pts/0        Mon Feb  6 12:17 - 15:11  (02:53)     :0.0
root     tty1         Mon Feb  6 12:12 - down   (02:58)
```

14. dmesg 命令

dmesg 命令用来显示系统的开机信息。由于 Linux 在开机时屏幕信息滚动很快，如果管理员没有查看仔细或者想详细查看登录时显示的各种信息，那么可以在开机后使用 dmesg 命令，实际上，开机信息已经储存在/var/log/dmesg 文件中，dmesg 就是通过读取这个文件再将信息显示出来的。其具体的使用格式为：

```
dmesg [选项]
```

dmesg 命令的选项及其具体含义如表 3-11 所示。

表 3-11 dmesg 命令选项及其含义

选 项	含 义
-c	显示开机信息后，清除缓存信息
-s	设置缓冲区大小，默认为 8192
-n	设置记录信息的层级

提示： Linux 在开机过程中，会将开机信息先储存在缓冲区(Ring Buffer)中，使用 dmesg 可以对这个缓冲区进行大小、清除等设置。

15. free 命令

free 命令用来显示系统内存的使用状况，包括物理内存、虚拟内存、共享内存和系统

缓存等。其具体的使用格式为：

```
free [选项] [-s 间隔秒数]
```

free 命令的可用选项及其具体含义如表 3-12 所示。

表 3-12　free 命令选项及其含义

选　项	含　义
-b	以 Byte 为单位显示内存的使用情况
-m	以 MB 为单位显示内存的使用情况
-K	以 KB 为单位显示内存的使用情况
-t	显示内存的总和
-s 间隔秒数	根据指定的间隔秒数持续显示内存的使用情况
-o	不显示系统的缓冲区列

例如，以 MB 为单位显示内存的使用情况，可以使用以下命令：

```
[root@CentOS ~]# free -m
              total       used        free     shared     buffers      cached
Mem:           1010        520         490          0          53         334
-/+ buffers/cache:          131         879
Swap:          2047          0        2047
```

从上面的显示信息我们可以看出，系统总内存大小为 1GB，已经使用了 520MB，而交换区(Swap)的大小为 2GB，还没有使用。

16. ps 命令

ps 命令用来显示系统当前的运行进程，其使用格式如下：

```
ps [选项]
```

ps 的功能非常强大，是管理员在日常维护中经常要用到的命令，使用该命令可以确定哪些进程正在运行，进程占用了多少资源，进程的运行状态是否正常，进程是否已经结束，有没有出现僵尸进程等。其可以使用的选项也非常多，在这里我们仅列出一些常用的选项，如表 3-13 所示。

表 3-13　ps 命令常用选项及其含义

选　项	含　义
-A 或者-e	显示所有进程
-a	显示一个终端的所有进程，包括含有每个程序的完整路径
-d	显示所有进程，但省略所有的会话引线
-x	显示所有系统进程，没有控制终端的进程，同时显示各个命令的具体路径。dx 不可合用
-p	pid 进程使用 CPU 的时间
-u uid 或者 username	选择有效的用户 id 或者是用户名

续表

选　项	含　义
-g gid 或者 groupname	显示组的所有进程
U username	显示该用户下的所有进程，且显示各个命令的详细路径，如：ps U zhang
-f	详细显示程序执行的路径群，通常和其他选项联用。如：ps -fa 或者 ps –fx
-l	长格式(有 F、wchan、C 等字段)
-j	作业格式
-o	用户自定义格式
v	以虚拟存储器格式显示
s	以信号格式显示
-m	显示所有的线程
e	命令之后显示环境(如：ps -d e; ps -a e)
h	不显示第一行
-c	只显示进程的名称，不显示进程的完成路径

直接使用 ps 命令可以查看使用者自己的进程，如下面的命令和显示内容：

```
[root@CentOS ~]# ps
  PID TTY        TIME  CMD
 5255 pts/0    00:00:00 bash
 5269 pts/0    00:00:00 ps
```

显示的项目主要有：

- PID：进程的标识号。
- TTY：进程所属的终端控制台。
- TIME：进程使用的总 CPU 时间。
- CMD：正在执行的命令行。

ps 命令常用的选项有 e、f、a、u 等，例如，如果要查看系统的所有进程，可以使用以下的命令：

```
[root@CentOS ~]# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 11:23 ?        00:00:01 init [3]
root         2     1  0 11:23 ?        00:00:00 [migration/0]
root         3     1  0 11:23 ?        00:00:00 [ksoftirqd/0]
root         4     1  0 11:23 ?        00:00:00 [watchdog/0]
root         5     1  0 11:23 ?        00:00:00 [events/0]
root         6     1  0 11:23 ?        00:00:00 [khelper]
…
```

17. top 命令

top 命令提供了实时监控系统处理器状态的功能，它能够实时显示出系统中各个进程的资源占用情况。该命令还可以按照 CPU 的使用、内存使用和执行时间等数据对系统任务进程进行排序显示，并且 top 命令还可以通过交互式的命令进行设定显示效果。

top 命令的使用格式为：

```
top [选项]
```

top 的选项也非常丰富，本书只列出一些常用的选项及其具体含义，如表 3-14 所示。

表 3-14　top 命令常用选项及其含义

选　项	含　义
-d	指定每两次屏幕信息刷新之间的时间间隔
-i	不显示闲置进程或者僵尸进程
-c	显示进程的整个命令路径，而不是只显示命令名称
-s	使 top 命令在安全模式下运行，为了避免潜在的不安全性，此时 top 的交互式命令不可用
-b	分屏显示输出信息，结合“-n”选项可以将屏幕信息输出到文档
-n	Top 输出信息的更新次数，完成后将退出 top 命令

与 ps 命令的最大不同在于，top 命令可以持续运行——即交互模式，可以实时显示系统的运行状态。在交互模式下，有很多与普通模式不同的命令，这些命令都是单个字母，从应用的角度来说，掌握这些命令至关重要。top 命令的交互模式命令及其具体含义如表 3-15 所示。

表 3-15　top 交互模式常用命令及其含义

命　令	含　义
H 或者?	显示帮助信息，给出交互式命令行的一些基本说明
K	终止一个进程，系统将提示用户输入需要终止进程的 PID
I	忽略/显示闲置进程或者僵尸进程
S	改变 top 输出信息两次刷新之间的时间。系统将提示输入新的刷新时间，单位为秒，也可以通过输入小数来设定为毫秒级；如果输入 0，那么系统输出将不断刷新，默认的刷新时间为 5 秒。如果管理员是通过肉眼进行观察，建议设置为 1~30 秒之间
o 或者 O	改变 top 输出信息中显示项目的排序。按小写的 a~z 键可以将相应的列向右移动，而按大写的 A~Z 键可以将相应的列向左移动，最后按 Enter 键确定
f 或者 F	从当前显示列表中添加或者删除项目。按 f 键后会在屏幕中显示出列的序号，以“a~z”编号，按相应的字母按钮可以选择显示或者隐藏该列，最后按 Enter 键确定
m	切换显示内存信息
t	切换显示进程和 CPU 状态信息
r	重新设置一个进程的优先级。系统将提示用户输入需要修改的进程 PID 以及需要设置的进程优先级。请注意，输入一个正值将降低其优先级；输入一个负值将提高其优先级，默认值是 10
l	切换显示平均负载和启动时间信息
q	退出 top 命令
c	切换显示完整命令行和命令名称信息

续表

命　令	含　义
M	根据使用内存的大小进行排序输出
P	根据 CPU 的使用百分比大小排序输出
T	根据累计时间进行排序输出
S	切换到累计模式
W	将当前 top 设置写入~/.toprc 文件中

如果管理员希望实时查看系统进程及资源的使用情况，可以直接输入 top 命令即可，屏幕将切换到 top 程序的显示界面，显示的内容如下所示：

```
top - 09:13:04 up 3 min,  2 users,  load average: 2.09, 0.86, 0.32
Tasks: 119 total,   2 running, 117 sleeping,   0 stopped,   0 zombie
Cpu(s): 10.8%us,  9.5%sy,  0.5%ni, 23.4%id, 55.1%wa,  0.1%hi,  0.7%si,  0.0%st
Mem:   1035108k total,   410288k used,   624820k free,    19400k buffers
Swap:  2097144k total,        0k used,  2097144k free,   268376k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 4584 root      35  19 59056  16m 6308 D  8.6  1.7   0:01.17 yum-updatesd-he
 4399 root      14  -1 35064 9508 5508 S  1.2  0.9   0:01.20 X
    1 root      15   0  2072  628  544 S  0.0  0.1   0:00.99 init
    2 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 migration/0
    3 root      34  19     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/0
    4 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
    5 root      10  -5     0    0    0 S  0.0  0.0   0:00.00 events/0
    6 root      10  -5     0    0    0 S  0.0  0.0   0:00.00 khelper
    7 root      10  -5     0    0    0 S  0.0  0.0   0:00.00 kthread
   10 root      10  -5     0    0    0 S  0.0  0.0   0:00.07 kblockd/0
   11 root      20  -5     0    0    0 S  0.0  0.0   0:00.00 kacpid
  175 root      17  -5     0    0    0 S  0.0  0.0   0:00.00 cqueue/0
  178 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 khubd
  180 root      12  -5     0    0    0 S  0.0  0.0   0:00.00 kseriod
  246 root      15   0     0    0    0 S  0.0  0.0   0:00.00 khungtaskd
  247 root      23   0     0    0    0 S  0.0  0.0   0:00.00 pdflush
  248 root      15   0     0    0    0 S  0.0  0.0   0:00.00 pdflush
```

从上面的内容我们可以看出，top 输出信息分为上半部分的统计信息区和下半部分的进程信息区两块，其中统计信息区为前 5 行所显示的信息。

1) 统计信息区

统计信息区主要分为 3 块内容，其中第 1 行为任务队列的信息，包含的信息有：

- 当前系统时间。
- 系统已经运行的时间。
- 当前登录系统的用户数。
- 系统的平均负载。

第 2 行和第 3 行为进程和 CPU 的信息。具体包括：

- 进程的总数。
- 正在运行的进程数。

- 处于休眠的进程数。
- 停止的进程数。
- 僵尸进程数。
- 用户进程占用 CPU 的百分比。
- 系统进程占用 CPU 的百分比。
- 用户进程空间内改变过优先级的进程占用 CPU 的百分比。
- 空闲 CPU 占用的百分比。
- 等待输入输出的进程占用 CPU 的百分比。

第 4 行和第 5 行为内存的基本信息，主要包括：

- 系统的物理内存大小。
- 已经使用的物理内存大小。
- 目前空闲内存的大小。
- 用作内核缓冲区的内存的大小。
- 交换分区内存的大小。
- 已经使用的交换分区的大小。
- 空闲的交换分区的大小。
- 高速缓存的大小。

2) 进程信息区

进程信息区主要显示了每个进程的实时运行状态，其中字段的具体含义为：

- PID：进程的 id。
- USER：进程所有者的用户名。
- PR：进程的优先级。
- NI：nice 值，负值表示高优先级，正值表示低优先级。
- VIRT：进程使用的虚拟内存总量，单位为 KB。
- RES：进程使用的物理内存的大小，单位 KB。
- SHR：共享内存的大小，单位 KB。
- S：进程状态，D 表示不可中断的休眠进程，R 表示运行状态，S 表示休眠状态，T 表示跟踪/停止，Z 表示僵尸进程。
- %CPU：上次更新到现在的 CPU 占用时间百分比。
- %MEM：进程占用物理内存的百分比。
- TIME+：进程使用的 CPU 时间总和，单位为 1/100 秒。
- COMMAND：正在运行进程的命令名或者命令路径。

3.2.2　文件管理类

文件管理类命令主要用于对 Linux 文件的操作和编辑。

1. mkdir 命令

mkdir 命令用来在创建目录。其使用格式为：

```
mkdir [选项] 目录名
```

mkdir 可使用的选项如表 3-16 所示。

表 3-16　mkdir 命令选项及其含义

选　项	含　义
-m	对新建的目录设置权限
-p	可以指定一个路径名。如果此路径中的某些目录不存在，系统将自动建立这些路径中的所有尚未建立的目录

使用“-p”选项，我们可以实现一个命令创建多个目录的目的。例如，如果我们要在/usr 目录下创建 usr1 目录，并且在 usr1 目录下再创建 usr2 目录。那么可以使用以下命令：

```
[root@CentOS /]# mkdir -p /usr/usr1/usr2
```

2. more 命令

more 命令主要用于文本显示，当一个文本文件的内容过多，无法在一个屏幕上完整显示时，就需要用到 more 命令。more 命令在读取文本文件时，每次只显示一屏的内容，然后就会暂定，在屏幕的底部显示“More”即为还有更多的内容。直到用户按下空格键，才会继续显示下一屏的内容，依此类推。more 命令的使用格式为：

```
more [选项] 文件名
```

more 命令的选项及其具体含义如表 3-17 所示。

表 3-17　more 命令选项及其含义

选　项	含　义
-d	在屏幕的底部显示帮助信息，包括继续按空格继续，按“q”退出等
-s	将输出文件中的多个空行显示为只有一个空行输出
-p	先清除显示屏幕以前的信息，再显示文本文件的信息
-c	显示文件时，每屏幕都先清除屏幕信息，然后再显示文本内容

例如：如果要用每 10 行一屏的形式显示“acpid”文件中的信息。可以使用以下命令：

```
[root@CentOS log]# more -10 acpid
[Wed Jan 18 23:24:46 2012] starting up
[Wed Jan 18 23:24:46 2012] 1 rule loaded
[Wed Jan 18 23:24:48 2012] client connected from 2984[68:68]
[Wed Jan 18 23:24:48 2012] 1 client rule loaded
[Wed Jan 18 23:28:18 2012] exiting
[Wed Jan 18 23:29:36 2012] starting up
[Wed Jan 18 23:29:36 2012] 1 rule loaded
[Wed Jan 18 23:29:37 2012] client connected from 5499[68:68]
[Wed Jan 18 23:29:37 2012] 1 client rule loaded
[Wed Jan 18 23:58:41 2012] client connected from 5990[0:0]
--More--(8%)
```

3. cat 命令

cat 命令用来将文件中的内容打印到标准输出中，其功能类似于 more 命令，但不同的是，cat 命令还可以用户合并文件。用户显示文件内容时，其使用格式如下：

```
cat [选项] 文件名
```

用于合并文件时，其使用格式为：

```
cat 文件 1 文件 2 > 文件 3
```

cat 命令可用的选项及其具体含义如表 3-18 所示。

表 3-18 cat 命令选项及其含义

选 项	含 义
-A	将文件中的 Tab 输出为“^I”，同时在每行的末尾显示“$”符号
-b	将文件中所有非空行按顺序编号

下面详细介绍一下 cat 用来合并文件的功能的使用方法。

我们已经在/home 目录下创建了两个文件 test1.txt 和 test2.txt，现在我们要将这两个文件中的内容合并到 test3.txt 中并输出。这一过程使用的命令及其显示内容如下所示：

```
[root@CentOS home]# ls
test1.txt  test2.txt
[root@CentOS home]# cat test1.txt
this is test1.txt file's content.
[root@CentOS home]# cat test2.txt
this is test2.txt file's content.
[root@CentOS home]# cat test1.txt test2.txt >test3.txt
[root@CentOS home]# ls
test1.txt  test2.txt  test3.txt
[root@CentOS home]# cat test3.txt
this is test1.txt file's content.
this is test2.txt file's content.
```

4. diff 命令

diff 命令用来比较文件的差异。当使用 diff 命令比较两个文件时，它将以逐行比较的方式显示两个文件的异同。而如果使用 diff 命令比较两个目录，那么它会比较两个目录中所有的同名文件，但子目录并不会再进行比较。diff 命令的使用格式为：

```
diff [选项] 文件 1 文件 2
```

diff 命令的可用选项及其具体含义如表 3-19 所示。

表 3-19 diff 命令选项及其含义

选 项	含 义
-c	显示全部内容，并标出不同之处
-b	忽略行尾的空格。同时字符串中的一个或者多个空格将视为相同内容
-r	当比较两个目录时可以使用此选项，意为要求比较其子目录中的文件的不同
-s	当两个文件相同时，显示文件相同的信息

例如，我们继续以上面的 test1.txt 和 test2.txt 文件为例，比较这两个文件异同的命令及显示内容如下：

```
[root@CentOS home]# diff test1.txt test2.txt
1c1
< this is test1.txt file's content.
---
> this is test2.txt file's content.
```

在上面比较结果中。“1c1”表示比较的是两个文件的第一行，其中“c”代表本行内容不同，另外，还有“a”代表“文件 2”中附加了内容，“d”代表“文件 2”中删除了内容。

下面的内容为显示两个文件中的不同行的内容，其中以“<”开始的行是“文件 1”中的内容，以“>”开始的行是“文件 2”中的内容。

为了能够表示的更清楚，下面我们对 test1.txt 和 test2.txt 做简单的修改，再次比较，读者就能看清其中所表达的含义了。显示内容如下所示：

```
[root@CentOS home]# cat test1.txt
this is test1.txt file's content.
this is the same line.
this is test1.txt addtion.

this is test1.txt addtion.this is test2.txt addtion.
[root@CentOS home]# cat test2.txt
this is test2.txt file's content.
this is the same line.

this is test2.txt addtion.
[root@CentOS home]# diff test1.txt test2.txt
1c1
< this is test1.txt file's content.
---
> this is test2.txt file's content.
3d2
< this is test1.txt addtion.
5c4
< this is test1.txt addtion.this is test2.txt addtion.
---
> this is test2.txt addtion.
```

5. grep 命令

grep 命令用来过滤文本中的内容，可以根据指定的字符串，对文本文件中的内容进行搜索，将含有指定字符串的行内容显示出来。其具体使用格式为：

```
grep [选项] 字符串 文件名
```

grep 命令的可用选项很多，这里我们只列出常用的选项及其含义，如表 3-20 所示。

表 3-20 grep 常用命令选项及其含义

选 项	含 义
-A n	出列列出符合条件的行外，还列出每个符合条件行的后 n 行
-c	只显示符合条件的行，而不显示每行的具体信息

续表

选　项	含　义
-f file	用户先将搜索的样式写入 file 文件中，然后根据这个文件中的条件进行搜索
-i	搜索时忽略大小写
-n	在显示的搜索结果上显示行号
-B n	与“-A”功能相反，列出符合条件行的前 n 行

例如，列出 acpid 文件中包含 load 字符串的行，忽略大小写，并且标识出每行的具体行号，其命令及显示内容为：

```
[root@CentOS log]# grep -ni load acpid
2:[Wed Jan 18 23:24:46 2012] 1 rule loaded
4:[Wed Jan 18 23:24:48 2012] 1 client rule loaded
7:[Wed Jan 18 23:29:36 2012] 1 rule loaded
9:[Wed Jan 18 23:29:37 2012] 1 client rule loaded
11:[Wed Jan 18 23:58:41 2012] 1 client rule loaded
14:[Thu Jan 19 00:10:37 2012] 1 client rule loaded
16:[Thu Jan 19 00:36:06 2012] 1 client rule loaded
20:[Thu Jan 19 00:36:17 2012] 1 client rule loaded
23:[Wed Jan 18 17:44:54 2012] 1 rule loaded
25:[Wed Jan 18 17:44:58 2012] 1 client rule loaded
27:[Wed Jan 18 17:45:23 2012] 1 client rule loaded
…
```

6. rm 命令

rm 命令用来删除文件或者目录，如果是链接文件，那么只能断开链接，原文件将保持不变。其使用格式如下：

```
rm [选项] 文件或者目录名
```

rm 命令的可用选项及其具体含义如表 3-21 所示。

表 3-21　rm 命令选项及其含义

选　项	含　义
-r	删除目录下的全部文件及其子目录中的全部文件。如果没有“-r”选项，将不删除目录
-f	忽略所有提示和选择，直接删除
-i	在删除每个文件之间询问用户

如果管理员已经确认系统中某个目录下的全部内容都需要删除，一般可以使用“rm -rf”命令来删除，但请读者注意，在使用 rm 命令时一定要谨慎，因为一旦删除文件或者目录，Linux 系统将无法恢复这些文件，所以，在不确定的情况下，可以使用“rm -i”来删除文件，此时系统每删除一个文件都会提示用户以确认是否删除。

7. touch 命令

touch 命令用来改变指定文件的访问时间和修改时间，如果文件不存在，touch 命令会首先创建此文件；如果没有指定时间，那么会直接使用当前时间。touch 命令的使用格

式为：

```
touch [选项] 设定时间 文件名
```

touch 命令的可用选项及其含义如表 3-22 所示。

表 3-22 touch 命令选项及其含义

选 项	含 义
-a	改变文件的访问时间为系统的当前时间。无需使用“设定时间”
-m	改变文件的修改时间为系统的当前时间。无需使用“设定时间”
-c	如果文件不存在，不创建此文件，也不给出提示
-d 或者-t	使用指定的日期时间
-r 参考文件或者目录	把指定文件或者目录的日期设定为参考文件或者目录的时间

下面，我们将以一个修改文件时间的小实例来介绍和回顾 ls、date 和 touch 命令的用法。

```
[root@CentOS home]# touch test.txt   #创建 test.txt 文件
[root@CentOS home]# ls -l           #显示文件创建时间
total 28
-rw-r--r-- 1 root root  138 Feb 10 12:27 test1.txt
-rw-r--r-- 1 root root    0 Feb 11 11:13 test.txt
[root@CentOS home]# ls -lu          #显示文件访问时间
total 28
-rw-r--r-- 1 root root  138 Feb 10 12:28 test1.txt
-rw-r--r-- 1 root root    0 Feb 11 11:13 test.txt
[root@CentOS home]# date      #查看系统当前时间
Sat Feb 11 11:18:18 CST 2012
[root@CentOS home]# touch -a test.txt  #改变访问时间为系统时间
[root@CentOS home]# ls -lu
total 28
-rw-r--r-- 1 root root  138 Feb 10 12:28 test1.txt
-rw-r--r-- 1 root root    0 Feb 11 11:18 test.txt
[root@CentOS home]# ls -l
total 28
-rw-r--r-- 1 root root  138 Feb 10 12:27 test1.txt
-rw-r--r-- 1 root root    0 Feb 11 11:13 test.txt
[root@CentOS home]# touch -m test.txt   #改变修改时间为系统时间
[root@CentOS home]# touch -a -r test1.txt test.txt  #将文件访问时间修改为
test1.txt 的访问时间
[root@CentOS home]# ls -lu
total 28
-rw-r--r-- 1 root root  138 Feb 10 12:28 test1.txt
-rw-r--r-- 1 root root    0 Feb 10 12:28 test.txt
[root@CentOS home]# touch -d "20120101 12:00" test.txt   #将 test.txt 文件
的修改时间设定为指定的时间
[root@CentOS home]# ls -l
total 28
-rw-r--r-- 1 root root  138 Feb 10 12:27 test1.txt
-rw-r--r-- 1 root root    0 Jan  1 12:00 test.txt
```

8. ln 命令

ln 命令为某一个文件或者目录在另外一个位置建立一个链接。当用户需要在不同的目

录用到相同的文件时，不需要在每一个需要的目录下都放一个相同的文件，只要在某个固定的目录放上该文件，然后在其他的目录下用 ln 命令链接(Link)它就可以了，这样就不必重复地占用磁盘空间。ln 命令的使用格式如下：

```
ln [选项] 源文件 目标链接
```

ln 命令可以使用的选项及其含义如表 3-23 所示。

表 3-23 ln 命令选项及其含义

选 项	含 义
-f	如果在目标位置存在于链接同名的文件，则删除这个文件
-s	进行软链接
-d	允许系统管理员硬链接自己的目录
-b	对将在链接时会被覆盖或者删除的文件进行备份

提示： 硬链接与软链接：Linux 下有两种链接：硬链接(Hard Link)和软链接(Symbolic Link，也称符号链接)。硬链接是指通过文件的索引节点来创建链接。在 Linux 文件系统中，保存在磁盘的所有文件都会分配一个编号，这个编号称为索引号(Inode Index)。多个文件可以指向同一个索引节点，这种多文件指向同一个索引节点的链接方法就叫做硬链接。硬链接意味着允许一个文件拥有多个有效的路径名，只要用户修改了任何一个路径中的文件，那么其他的链接文件也就随之更改，如果要删除硬链接文件，那么必须删除其所有链接，此文件的数据块才被释放。而软链接类似于 Windows 中的快捷方式，是一个指向真正文件或者目录位置的符号链接。

例如，下面的例子，我们首先在/home 目录下创建 linkdir 目录，然后在此目录中创建 test1.txt 文件的硬链接。可以看到，两个文件是完全相同的。

```
[root@CentOS home]# mkdir linkdir
[root@CentOS home]# ls
linkdir  lost+found  test1.txt  test2.txt  test3.txt  test.txt
[root@CentOS home]# ln test1.txt /home/linkdir
[root@CentOS home]# ll /home/test1.txt
-rw-r--r-- 2 root root 138 Feb 10 12:27 /home/test1.txt
[root@CentOS home]# ll /home/linkdir/test1.txt
-rw-r--r-- 2 root root 138 Feb 10 12:27 /home/linkdir/test1.txt
```

而下面的例子给出了在 linkdir 目录中创建 test2.txt 的软链接命令。

```
[root@CentOS home]# ln -s test2.txt /home/linkdir
[root@CentOS home]# ll /home/linkdir/test2.txt
lrwxrwxrwx 1 root root 9 Feb 11 12:17 /home/linkdir/test2.txt ->
test2.txt
```

9. file 命令

file 命令用来显示文件的类型。File 命令能识别的文件类型有目录、Shell 脚本、英文文本、二进制可执行文件、C 语言源文件、文本文件、DOS 的可执行文件。其使用格式

如下：

```
file [选项] 文件名
```

file 命令的选项及其具体含义如表 3-24 所示。

表 3-24　file 命令选项及其含义

选　项	含　义
-b	显示文件类型的结果，不显示对应文件名
-L	直接显示软链接所指向文件的类型
-z	显示压缩文件的信息
-I	如果文件不是常规文件，则不进一步对文件类型进行分类

下面的例子是几种常见类型文件的 file 属性。

```
[root@CentOS /]# file /etc/init.d/acpid
/etc/init.d/acpid: Bourne-Again shell script text executable     #shell
脚本执行文件
[root@CentOS /]# file /bin/sh
/bin/sh: symbolic link to `bash'      #软链接文件
[root@CentOS /]# file /home/test.txt
/home/test.txt: empty             #空文件
[root@CentOS /]# file /home/test1.txt
/home/test1.txt: ASCII English text    #ASCII 文本文件
[root@CentOS /]# file /dev/sda1
/dev/sda1: block special (8/1)        #设备文件
```

10. cp 命令

cp 命令是 Linux 下最常用的复制命令，用来复制文件或者目录，其功能与 Windows 下的 copy 命令非常类似，但功能更加强大。其使用格式为：

```
cp [选项] 源文件 目标文件
```

cp 的可用选项非常丰富，表 3-25 只列出了一些常用的选项。

表 3-25　cp 命令常用选项及其含义

选　项	含　义
-a	在复制目录时保留所有信息，包括文件链接文件属性、并复制子目录
-r	递归的复制源文件目录中的所有子目录中的文件。此时目标文件也必须是一个目录名
-d	复制是保留链接文件
-p	保留文件的修改日期和存取权限
-i	如果已经有相同的文件名的目标文件，则提示用户是否覆盖
-f	如果已经有相同的文件名的目标文件，则直接覆盖此文件

例如，将当前目录下的所有文件复制到/home/linkdir 目录中，则可以使用以下命令：

```
[root@CentOS home]# cp -r ./* /home/linkdir
```

11. find 命令

find 命令用来在至顶端的路径下查找文件。其使用格式为：

```
find 路径 [选项]
```

find 命令可用选项非常丰富，具体内容及含义如表 3-26 所示。

表 3-26　find 命令选项及其含义

选　项	含　义
-name 字符串	查找文件名符合字符串的所有文件。其中字符串可以使用通配符
-lname 字符串	查找文件名符合字符创的所有链接文件。其中字符串可以使用通配符
-gid n	查找属于 ID 号为 n 的用户组的所有文件
-uid n	查找属于 ID 号为 n 的用户的所有文件
-empty	查找空文件和目录
-path 字符串	查找路径名符合字符串的所有文件。其中字符串可以使用通配符
-group 用户组名	查找属于此用户组名的所有文件
-depth	递归查找目录及其子目录中符合条件的所有文件
-prune 目录名	搜索时不搜索此目录名下的文件。与“-depth”命令冲突
-size n	查找文件长度为 n 的文件
-user 用户名	查找属于此用户的所有文件
-mtime +n 或者-n	按时间搜索，“+n”表示 n 天之前，“-n”代表今天到 n 天前之间的文件
-type 文件类型	按指定文件类型搜索。文件类型包括 b(块设备文件)、c(字符设备文件)、f(普通文件)、l(符号文件)、d(目录)、p(管道)、s(socket 文件)
-print	将搜索结果输出到标准输出
-exec 命令 {} \;	对符合搜索条件的文件执行给出的命令
-ok 命令 {} \;	对符合搜索条件的文件执行给出的命令，但执行前会询问用户

例如，查找在用户 user1 的所有普通文件中查找，这些文件是两天以前的，并且查找时不搜索/usr/bin 目录，文件名为 main.c，最后将结果输出到屏幕，并且删除所有搜索到的文件，此命令可以写成如下格式：

```
[root@CentOS home]#find / -path "usr/bin" -prune -o -name "main.c" -user
user1 -type f -mtime +2 -print -exec rm {} \;
```

12. split 命令

split 命令用来将一个大文件分割为若干个小文件。其使用格式为：

```
split [选项] 待分割文件 分割后文件基础名
```

其中，“分割后文件基础名”的命令规则为：以“分割后文件基础名+序列长度英文字母(默认以 aa、ab、ac 等依次排序)”的形式逐个命名，如果没有指定“分割后文件基础名”，那么将默认以英文字母序列作为输出文件名。

split 命令的可用选项及其含义如表 3-27 所示。

表 3-27 split 命令选项及其含义

选 项	含 义
-b size	指定分割文件的大小，size 可以带有大小单位，如 b、KB、MB 等
-n	设定分割文件的长度，缺省为 1000 行
-d	将生成的文件序列以数字形式表示
-a	指定 split 命令生成的文件序列长度，默认为 2

例如，将 busybox 文件分割为每个 100K 大小的文件，可以使用以下命名。

```
[root@CentOS home]# ls -l
total 2068
-rwxr-xr-x 1 root root 2090944 Feb 12 22:46 busybox
drwxr-xr-x 4 root root    4096 Feb 11 17:27 linkdir
drwx------ 2 root root   16384 Jan 18 23:07 lost+found
[root@CentOS home]# split -b 100k busybox busybox
[root@CentOS home]# ls
busybox    busyboxad  busyboxah  busyboxal  busyboxap  busyboxat
busyboxaa  busyboxae  busyboxai  busyboxam  busyboxaq  busyboxau
busyboxab  busyboxaf  busyboxaj  busyboxan  busyboxar  linkdir
busyboxac  busyboxag  busyboxak  busyboxao  busyboxas  lost+found
```

13. mv 命令

mv 命令用来移动文件或者文件夹。不仅如此，当 mv 的源和目标同时为文件或者目录时，还可以完成文件或者目录的重命名功能。其使用格式为：

```
mv [选项] 源文件 目标文件
```

mv 命令的可用选项及其含义如表 3-28 所示。

表 3-28 mv 命令选项及其含义

选 项	含 义
-i	交互式操作。对已经存在的文件或者目录，系统会询问是否覆盖
-f	禁止交互式操作，mv 命令将直接覆盖原有的文件或者目录，不做任何提示

例如下面的例子，我们选创建一个目录 mvdir 以及文件 mvfile，然后将 mvfile 文件移动到 mvdir 中，再修改 mvfile 文件名为 mvfile1。

```
[root@CentOS home]# mkdir mvdir    #创建目录
[root@CentOS home]# touch mvfile    #创建文件
[root@CentOS home]# ls
lost+found  mvdir  mvfile
[root@CentOS home]# mv mvfile mvdir  #将文件移动到目录中
[root@CentOS home]# cd mvdir
[root@CentOS mvdir]# ls
mvfile
[root@CentOS mvdir]# mv mvfile mvfile1   #修改文件名
[root@CentOS mvdir]# ls
mvfile1
```

3.2.3　压缩类

不管是为了存档，还是为了节约空间，对于系统管理员来说，压缩文件是经常要用到的功能。与 Windows 相比，Linux 提供了更加丰富的压缩和解压缩命令和功能。

1. zip/unzip 命令

zip 是最著名和拥有优秀压缩算法的压缩程序，在 Windows 中应用广泛，在 Linux 也同样可以使用。zip 命令可以将文件或者目录进行压缩，生成以“.zip”为后缀的压缩包。其具体的使用格式如下：

```
zip [选项] 压缩文件名 待压缩文件列表
unzip [选项] 压缩文件名
```

zip 命令可用选项及其含义如表 3-29 所示。

表 3-29　zip 命令选项及其含义

选　项	含　义
-r	压缩指定目录及其子目录的全部文件
-d	从压缩文件内删除文件
-I 文件列表	只压缩文件列表中的文件
-x 文件列表	压缩时不压缩文件列表中的文件
-u	更新文件到压缩文件中
-m	将文件移动到压缩文件中
-F	修复损坏的压缩文件
-T	检测压缩文件的完好性
-(1~9)	用指定的数字设置压缩比，1 为最低压缩比，但速度最快；9 为最高压缩比，但速度最慢；系统默认压缩比为 6

unzip 命令可用选项及其含义如表 3-30 所示。

表 3-30　unzip 命令选项及其含义

选　项	含　义
-x 文件列表	不解压文件列表中的文件
-t	测试压缩文件的完好性
-v	查看压缩文件的详细信息，包括压缩文件中的文件大小、压缩比等信息
-n	解压时不覆盖已存在的文件
-o	解压时覆盖已存在的文件，且不提示
-d 目录名	了解压文件到指定的目录中

下面的例子给出了压缩文件、解压文件，删除压缩文件中内容的具体操作方法。

```
[root@CentOS /]# zip -9r /var/log/log.zip /var/log     #压缩文件，使用最高压
缩比，保存为 log.zip
```

```
  adding: var/log/ (stored 0%)
  adding: var/log/rpmpkgs (deflated 71%)
  adding: var/log/rpmpkgs.3 (deflated 71%)
  adding: var/log/cron (deflated 83%)
  adding: var/log/messages.1 (deflated 79%)
…
[root@CentOS /]# zip /var/log/log.zip -d var/log/rpmpkgs.3  #删除压缩包中的 rpmpkgs.3 文件。
deleting: var/log/rpmpkgs.3
[root@CentOS /]# unzip -n /var/log/log.zip -d /var/log     #解压文件到指定目录，并且不覆盖原有文件。
Archive:  /var/log/log.zip
   creating: /var/log/var/log/
  inflating: /var/log/var/log/rpmpkgs
  inflating: /var/log/var/log/cron
  inflating: /var/log/var/log/messages.1
  inflating: /var/log/var/log/cron.3
…
```

2. gzip 命令

gzip 命令可以对文件进行压缩和解压缩，压缩文件扩展名为“.gz”。与 zip 不同的是，gzip 只能压缩文件，不能压缩目录。其使用格式为：

```
gzip [选项] 压缩(解压缩)文件名
```

gzip 命令的选项及其含义如表 3-31 所示。

表 3-31 gzip 命令选项及其含义

选 项	含 义
-d	解压缩文件
-r	压缩指定目录及其子目录中的所有文件
-t	检查压缩文件的完整性
-v	对每个压缩和解压缩文件，显示其文件名和压缩比
-l	显示压缩文件的信息，包括压缩文件的大小，未压缩文档的大小，压缩比一级未压缩文档的名称等
- (1~9)	用指定的数字设置压缩比，与 zip 命令相同

3. bzip2 命令

bzip2 的功能与 gzip 非常类似，智能对文件进行压缩，生成以“.bz2”为后缀的压缩文件，而在压缩后的文件中不能带有目录。bzip2 的使用格式如下：

```
bzip2 [选项] 待压缩或者解压缩文件
```

bzip2 命令的选项及其含义如表 3-32 所示。

注意： 与 gzip 不同，bzip2 压缩文件以及解压缩文件后，会将源文件删除。

表 3-32　bzip2 命令选项及其含义

选　项	含　义
-d	执行解压缩
-v	压缩或者解压缩文件时，显示详细信息
-k	压缩或者解压缩文件后，保留原始文件
-f	遇到已有文件时，直接覆盖，而不提示
-t	测试压缩文件的完整性
-(1~9)	用指定的数字设置压缩比，与 zip 命令含义相同

4. tar 命令

tar 命令是 Linux 中经常用到的文档备份工具，它的功能是对文件和目录进行打包归档，形成一个文件，但并不进行压缩。其使用的具体格式为：

```
tar [主选项+辅助选项] 文件或者目录
```

tar 命令可用的选项很多，又可以分为主选项和辅助选项，经常用到的主选项如表 3-33 所示。

表 3-33　tar 命令主选项及其含义

选　项	含　义
-c	创建新的文件
-r	将文件添加到 tar 文件的末尾
-t	列出 tar 文件中的文件列表
-x	从 tar 文件中还原文件
-u	更新 tar 文件，用新的文件替换 tar 文件中的同名文件

tar 命令常用的辅助选项及其含义如表 3-34 所示。

表 3-34　tar 命令辅助选项及其含义

选　项	含　义
-z	调用 gzip 命令对归档文件进行压缩或者解压缩
-w	在还原文件时，将所有文件的修改日期改为现在时间
-j	调用 bzip2 命令对归档文件进行压缩或者解压缩
-f	指定 tar 文件存储的设备类型，默认是磁盘，需要指定档案文件名；如果是磁带，那么需要指定磁带设备名。这里读者必须注意，“-f”必须为 tar 命令的最后一个选项，也就是说，“-f”后不能再有任何其他选项
-v	在创建 tar 文件的过程中，显示文档的名称
-p	在创建 tar 文件的过程中，保持所有文件属性不发生改变
-N “yyyy/mm/dd”	只将给出日期时间之后的文件放入 tar 文档中
--exclude file	不讲指定文件放入 tar 文档中

例如，我们要对/var/log 目录下的所有文件进行打包并且压缩，还要显示打包的详细信息，可以使用以下两个命令中的一个：

```
[root@CentOS /]# tar -zcvf /home/varlog.tar.gz /var/log  #使用 gzip 命令压缩
[root@CentOS /]# tar -jcvf /home/varlog.tar.bz2 /var/log  #使用 bzip2 命令压缩
```

从上面的两条命令我们可以看出，使用 tar 命令生成的打包文件对文件名的后缀并没有要求，用户可以自由的命名。但在实际的使用中，为了便于辨识和传播，在命名时还是有一些习惯性的规则，这些规则已经被广泛使用并且达成了共识，推荐读者在使用 tar 命令对程序或者文档进行打包压缩时也遵循这些规则。

- 规则一：所有使用 tar 命令打包的文件应该以“.tar”为文件的后缀。
- 规则二：如果在打包的过程中使用了 gzip 命令进行压缩，那么应该将后缀命名为“.tar.gz”。
- 规则三：如果在打包过程中使用了 bzip2 命令进行压缩，那么应该将后缀命名为“.tar.bz2”。

5. dd 命令

dd 命令可以用来转换或者复制文件，同时也可以对设备进行备份。其使用格式为：

```
dd if="输入文件名" of="输出文件名" bs="每块大小" count="块数量"
```

dd 命令各选项的具体含义如表 3-35 所示。

表 3-35　dd 命令选项及其含义

参　数	含　义
if	输入文件名，可以是设备，例如磁盘的某个分区
of	输出文件名，可以是磁盘或者磁带等
bs	指定的块的大小，默认为 512bytes
count	生成的块的数量

例如，我们将/etv/inittab 被分到/opt/inittab.bak 文件中，可以使用以下的命令：

```
[root@CentOS ~]# dd if=/etc/inittab of=/opt/inittab.bak
3+1 records in
3+1 records out
1666 bytes (1.7 kB) copied, 0.0180618 seconds, 92.2 kB/s
[root@CentOS ~]# ls -al /etc/inittab /opt/inittab.bak
-rw-r--r-- 1 root root 1666 Jan 18 23:22 /etc/inittab
-rw-r--r-- 1 root root 1666 Feb 15 13:57 /opt/inittab.bak
```

从上面的实例我们可以看出，源文件和备份文件没有任何差异。dd 命令除了可以进行常规的文件备份外，还可以备份磁盘分区。例如下面的实例，我们将 sda1 分区被分到/opt 目录中。

```
[root@CentOS ~]# dd if=/dev/sda1 of=/opt/sda1.bak
```

注意： 在备份磁盘分区时，一定要注意输出路径“of”不能处于备份的磁盘分区中，否则将造成死循环，永远无法备份完成。而且在恢复磁盘备份时，最好保证恢复目标分区与备份分区大小相同，否则，如果目标分区大于备份分区，则会造成多余的空间无法利用；而如果目标分区小于备份分区，会提示磁盘空间不足。

6. cpio 命令

cpio 命令可以通过重定向的方式将文件进行备份和还原。它可以解压以“.cpio”或者“.tar”为后缀的文件。其使用格式为：

```
cpio [选项] <(或者>) 文件或者设备名
```

cpio 可用的选项及其含义如表 3-36 所示。

表 3-36 cpio 命令选项及其含义

选　项	含　义
-o	将文件打包或者输出到指定的设备上
-i	将打包文件解压或者将设备上的备份还原到系统中
-t	查看 cpio 打包的文件内容或者输出到设备上的文件内容
-v	显示打包过程的文件名
-d	在 cpio 还原文件的过程中，自动建立对应的目录
-c	使用一种较新的存储方式
-B	将默认备份块设置为 5120bytes，而不是默认的 512bytes

cpio 命令还有一个特点，所有的备份文件必须指定出完整的路径以及文件名，而无法自动搜索或者备份某目录下的全部文件，所以，在实际的使用过程中，cpio 命令经常与 find 命令一起使用，例如下面的例子。

```
[root@CentOS ~]# find /etc -type f | cpio -ccvB >/home/etc.cpio
```

3.2.4 磁盘管理类

Linux 中的磁盘管理类命令的主要功能是查看磁盘空间的使用情况，磁盘问题的修复以及磁盘和内存的存储控制等。

1. df 命令

df 命令用来查看磁盘空间占用情况，其使用格式如下：

```
df [选项]
```

df 命令可用的选项及其含义如表 3-37 所示。

表 3-37　df 命令选项及其含义

选　项	含　义
-h	以更人性化的显示方式输出文件分区空间的使用情况
-k	以 kB 为单位输出文件系统分区占用情况
-m	以 MB 为单位输出文件系统分区占用情况
-a	列出所有文件系统的分区，包括 0 大小的文件系统分区
-i	列出文件系统分区的 innodes 信息
-T	显示磁盘分区的文件系统类型

例如，我们经常会查看当前系统分区的使用情况，下面就是以更加人性化的形式显示系统磁盘空间和文件系统类型的命令。

```
[root@CentOS ~]# df -hT
Filesystem    Type    Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogRoot
             ext3     16G  3.8G   11G  26% /
/dev/mapper/VolGroup00-Home
             ext3    1.9G   35M  1.8G   2% /home
/dev/sda1      ext3     99M   12M   82M  13% /boot
tmpfs        tmpfs    506M     0  506M   0% /dev/shm
```

2. du 命令

du 命令用来显示文件或者目录占用的磁盘空间的情况，使用格式如下：

```
du [选项] 文件或者目录
```

du 命令可用的选项及其具体含义如表 3-38 所示。

表 3-38　du 命令选项及其含义

选　项	含　义
-s	显示文件或者目录的大小，单位为 KB
-b	显示文件或者目录的大小，单位为 byte
-sh	以更人性化的方式显示文件或者目录的大小
-sm	显示文件或者目录的大小，单位为 MB

下面的实例给出了使用 du 命令查看文件大小的样式。

```
[root@CentOS usr]# du -sh ./*
170M    ./bin
8.0K    ./etc
8.0K    ./games
1.2M    ./include
1.9M    ./kerberos
1.6G    ./lib
23M     ./libexec
248K    ./local
34M     ./sbin
1.6G    ./share
```

```
24K     ./src
4.0K    ./tmp
24K     ./X11R6
```

3. fsck 命令

fsck 命令用来检查并且修复文件系统中的错误，使用格式如下：

```
fsck [选项] [设备名]
```

fsck 命令的可用选项及其具体含义如表 3-39 所示。

表 3-39　fsck 命令选项及其含义

选　项	含　义
-a	自动修复文件系统问题，没有提示
-r	采用互动的修复方式
-A	使用/etc/fstab 配置文件中的内容，检查文件内所列出的所有文件系统
-T	执行 fsck 命令时，不提示标题信息
-C	显示 fsck 命令的执行过程
-N	不执行命令，只列出实际执行会进行的操作
-t 文件系统类型	指出要检查的文件系统的具体类型

下面的实例给出了：

```
[root@CentOS ~]# fsck -srV -t ext3 /dev/sda1
fsck 1.39 (29-May-2006)
[/sbin/fsck.ext3 (1) -- /boot] fsck.ext3 -r /dev/sda1
e2fsck 1.39 (29-May-2006)
/boot: clean, 35/26104 files, 15577/104388 blocks
```

特别需要注意的是，在使用 fsck 命令修复某个文件系统之前，一定要确认此文件系统处于卸载状态，因为磁盘分区在挂载状态进行修复是极其不安全的，其中的数据很有可能会遭到破坏，更甚者甚至损坏整个磁盘。如果在挂载状态监测某个磁盘时，会给出以下的提示：

```
[root@CentOS ~]# fsck /dev/mapper/VolGroup00-LogRoot
fsck 1.39 (29-May-2006)
e2fsck 1.39 (29-May-2006)
/dev/mapper/VolGroup00-LogRoot is mounted.

WARNING!!! Running e2fsck on a mounted filesystem may cause
SEVERE filesystem damage.

Do you really want to continue (y/n)?
```

如果由于操作不当导致使用 fsck 命令修复的过程中有文件丢失，可以到对应的“lost+found”目录中寻找，此时，通过文件名往往已经无法分辨文件，管理员可以使用 file 命令来查看文件系统的类型，从而判断哪些是我们需要的文件。

4. sync 命令

sync 命令用来将内存中的数据写如入硬盘。在 Linux 系统中，对系统的修改等操作并

不会立刻写入硬盘中，而是先缓存在内存中，等到适当的时候再写入硬盘，这样主要是为了提高操作系统的运行效率。Linux 系统默认情况下每 3 秒自动运行一次 sync 操作，系统关机时也会执行 sync 命令，如果非正常关机，系统内存中的数据来不及写入硬盘，从而造成数据的丢失或者文件的损坏。sync 命令的使用非常简单，不需要参数，格式如下：

```
sync
```

5. eject 命令

eject 命令可以退出各种临时设备，例如光驱或者磁带。如果该设备已经挂载，则 eject 首先卸载设备，然后再退出设备。其使用格式如下：

```
eject [选项] 设备名
```

eject 命令可用的选项及其含义如表 3-40 所示。

表 3-40 eject 命令选项及其含义

选 项	含 义
-c 光驱编号	退出光驱，如果系统中有多个光驱，需要指定光驱编号
-d 或者—default	显示默认的设备，而不是实际执行操作
-f 或者--floppy	退出软盘
-q 或者--tape	退出磁带设备
-r 或者--cdrom	退出光盘
-t 或者--trayclose	关闭光驱的托盘
-n 或者-noop	显示指定的设备名对应的设备文件路径，默认为显示光驱的文件路径

例如，要显示光驱的文件路径，可以使用以下的命令。

```
[root@CentOS ~]# eject -n cdrom
eject: device is '/dev/hda'
```

6. mount/umount 命令

mount 和 umount 命令是 Linux 下挂载设备的常用命令，例如光驱、Windows 文件格式的磁盘等。mount 的优势在于支持多种文件系统类型，而且 mount 还能自动检测需要挂载的文件系统类型。其使用格式为：

```
mount [选项] [-t 文件类型] [-o options] device dir
umount dir
```

mount 命令的常用选项及其含义如表 3-41 所示。

表 3-41 mount 命令选项及其含义

选 项	含 义
-a	加载文件/etv/fstab 中的所有设备
-n	不将加载信息记录在/etc/fstab 文件中
-r	以只读方式加载设备

续表

选　项	含　义
-w	mount 的默认设置，以可写方式加载设备
-f 或-v	查看 mount 的挂载状态

在命令格式中 “-t 文件类型”中可以指定所要加载设备的文件类型。常用的文件类型有：

- 光盘或光盘镜像：iso9660。
- DOS fat16 文件系统：msdos。
- Windows 9x fat32 文件系统：vfat。
- Windows NT ntfs 文件系统：ntfs。
- Mount Windows 文件网络共享：smbfs。
- UNIX(LINUX) 文件网络共享：nfs。
- Linux 常用的文件系统：ext3/ext2。

而“-o options”则主要用来描述设备的挂载方式，常用的参数如表 3-42 所示。

表 3-42　-o 选项常用参数及其含义

参　数	含　义
loop	用来把一个文件当成硬盘分区挂接到系统中
ro	采用只读方式挂接设备
rw	采用读写方式挂接设备
iocharset	指定访问文件系统所用字符集
remount	重新加载设备，通常用于改变设备的设置状态
sync	以同步的方式执行文件系统的输入输出操作
user	可以让一般用户加载设备
default	使用默认设置加载设备

“device”指挂载的设备名，例如/dev/sda1、/dev/hda1 等。

“dir”指挂载到的位置，即挂载点，通常为 Linux 下的某个目录。

mount 的另一个优势是能够直接挂载光盘镜像，以实现类似光驱读取光盘的形式使用光盘镜像文件，例如下面的命令：

```
[root@CentOS ~]# mount -t iso9660 -o loop /iso/Cent5.5,iso /ixdba
```

此命令中，“loop”即指将光盘镜像文件 Cent5.5.iso 作为硬盘分区挂载到系统中，“/ixdba”为挂载点，挂载成功后，就可以通过访问/ixdba 目录来访问镜像文件中的内容了。

又如，我们可以将 Windows 操作系统中的硬盘分区挂载到 Linux 系统中进行使用，如下面的命令所示：

```
[root@CentOS ~]# mount -t vfat -o codepage=936,iocharest=cp936 /dev/hda3
/mnt/d
```

从上面的命令我们可以看出，首先需要指明挂载的分区格式为 vfat(即 FAT 格式)，而“-o”后的选项则是指明此挂载分区需要支持中文显示；“/dev/hda3”为 Windows 分区在 Linux 下的硬件标识；“/mnt/d”为挂载点。

另外，mount 命令还可以挂载其他 Linux 系统在网络中共享的分区，如下面的命令所示：

```
[root@CentOS ~]# mount -t nfs -o ro 192.168.1.2:/home/ mnt/nfs
```

在此命令中，挂载的文件系统为 nfs(即 Linux 文件网络共享)；“-o ro”指明以只读的方式挂载；192.168.1.2:/home/即网络共享的位置；“mnt/nfs”为挂载点。

3.2.5 网络配置类

网络配置类命令主要用于对 Linux 操作系统网络的设置及维护。

1. ifconfig 命令

ifconfig 命令用来配置网络或者显示当前网络接口的状态。其使用格式为：

```
ifconfig [选项] [网络接口] [指定操作]
```

ifconfig 命令的可用选项及其含义如表 3-43 所示。

表 3-43 ifconfig 命令选项及其含义

选项	含义
-a	显示所有网络接口的信息
-s	仅显示接口的摘要状态
-v	如果发现某网络接口出现错误，则返回错误信息，主要用户网络故障的诊断

命令格式中的“网络接口”指 Linux 下的网络接口名，通常以“eth0”、“lo”等形式表示，此选项可以不写，将显示所有网络设备的信息。

命令格式中的“指定操作”可以给出需要对网络接口进行的操作，可用操作包括以下几种：

- up：激活网络接口。
- down：与 up 相反，关闭某网络接口。
- netmask：为一个指定的网络接口设置子网掩码。
- addr：为网络接口设置 IP 地址。
- broadcast：为指定的网络接口设置广播地址。

下面的例子给出了显示所有网络接口的命令及其显示内容。

```
[root@CentOS ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:02:50:58
          inet addr:192.168.1.102  Bcast:255.255.255.255
Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe02:5058/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:595 errors:0 dropped:0 overruns:0 frame:0
          TX packets:441 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:1000
          RX bytes:728153 (711.0 KiB)  TX bytes:34271 (33.4 KiB)
          Interrupt:67 Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:37 errors:0 dropped:0 overruns:0 frame:0
          TX packets:37 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6066 (5.9 KiB)  TX bytes:6066 (5.9 KiB)
```

从上面的显示内容我们可以看出，使用 ifconfig 命令可以查看网卡的 MAC 地址信息、IP 地址信息、子网掩码、传输字节等。另外，“lo 网卡”是指回环网卡。

如果要设置网卡的 IP 地址，可以使用以下的命令：

```
[root@CentOS ~]# ifconfig eth0 192.168.2.102 netmask 255.255.255.0  #设置
第 1 个 IP 地址
[root@CentOS ~]# ifconfig eth0:0 192.168.3.102 netmask 255.255.255.0  #
设置第 2 个 IP 地址
```

从上面的例子我们可以看出，如果要给同一块网卡设置多个 IP 地址，可以使用 eth0:0、eth0:1 的书写方法。

使用 ipconfig 命令启用和关闭网卡的命令如下：

```
[root@CentOS ~]# ifconfig eth0 up
[root@CentOS ~]# ifconfig eth0 down
```

值得注意的是，使用 ipconfig 命令配置的网卡信息，在网卡重启或者系统重启后，所有的配置都会失效，如果需要使配置信息永久生效，需要修改网卡的配置文件。修改网卡配置文件的方法我们将在下节中介绍。

2. scp 命令

scp 命令主要用户在 Linux 网络中传输文件或者目录，其特点在于 scp 数据是使用 SSH 协议传输的，保证了数据传输的安全性。这也正是 scp(secure copy)命令的由来。scp 命令的使用格式如下：

```
scp [选项] 本地 Linux 系统文件路径 远程用户名@ip 地址:远程文件绝对路径
```

上面的格式是将本地文件传输到远程 Linux 系统中的命令格式，如果希望从远程计算机中复制文件，只需要将目标地址和源地址交换位置即可。

scp 命令可用的选项及其具体含义如表 3-44 所示。

表 3-44　scp 命令选项及其含义

选　项	含　义
-a	尽可能将档案状态、权限等资料都照原状予以复制
-r	若 source 中含有目录名，则将目录下之档案亦皆依序复制至目的地
-f	若目的地已经有相同档名的档案存在，则在复制前先予以删除再行复制

如果要将本地的文件复制到远程计算机中，可以使用如下格式的命令：

```
[root@CentOS ~]# scp /home/daisy/full.tar.gz root@172.19.2.75:/home/root
```

然后会提示用户输入远程计算机 172.19.2.75 主机的 root 用户的登录密码，接着就开始复制数据。

如果想反过来操作，把文件从远程主机复制到当前系统，也很简单。只需要将源和目标的位置对换即可。如下面的命令所示：

```
[root@CentOS ~]#scp root@/full.tar.gz 172.19.2.75:/home/root/full.tar.gz
home/daisy/full.tar.gz
```

3. netstat 命令

netstat 命令用来显示本机的网络状态信息。其使用格式如下：

```
netstat [选项]
```

netstat 命令的可用选项及其含义如表 3-45 所示。

表 3-45 netstat 命令选项及其含义

选 项	含 义
-a	显示本机所有连接和监听端口
-n	以 IP 地址的形式显示当前建立的网络连接和端口
-r	显示路由表信息
-s	显示按协议的统计信息。默认的统计协议有：IP、ICMP、TCP、UDP 的统计信息
-v	显示当前的有效连接
-t	显示所有 TCP 连接信息
-u	显示所有 UDP 连接信息
-c 秒数	设置刷新信息的时间
-i	显示自动配置接口的状态
-l	仅显示处于监听状态的网络接口
-p	显示连接对应的 PID 与程序名

例如，显示当前系统所有连接状态的命令及显示内容如下：

```
[root@CentOS ~]# netstat -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address             Foreign Address
State
tcp        0      0 CentOS:2208          *:*                    LISTEN
tcp        0      0 *:935              *:*                    LISTEN
tcp        0      0 *:sunrpc             *:*                    LISTEN
tcp        0      0 *:x11              *:*                    LISTEN
tcp        0      0 CentOS:ipp            *:*                    LISTEN
…
udp        0      0 *:43296                 *:*
udp        0      0 *:929                  *:*
udp        0      0 *:932                  *:*
udp        0      0 *:bootpc                 *:*
```

```
udp        0      0 *:mdns                      *:*
udp        0      0 *:sunrpc                    *:*
udp        0      0 *:ipp                       *:*
udp        0      0 *:35982                     *:*
udp        0      0 *:mdns                      *:*
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node Path
unix  2      [ ACC ]     STREAM     LISTENING     16648  /tmp/mapping-root
unix  2      [ ACC ]     STREAM     LISTENING     8696
/var/run/cups/cups.sock
unix  2      [ ACC ]     STREAM     LISTENING     12997  @/tmp/fam-root-
unix  2      [ ACC ]     STREAM     LISTENING     12634  /tmp/.font-unix/fs7100
unix  2      [ ACC ]     STREAM     LISTENING     15311  /tmp/.X11-unix/X0
…
```

从上面的例子我们可以看出，“netstat -a”命令主要显示了以下三方面的内容：

- 所有的 TCP 连接信息。
- 所有的 UDP 连接信息。
- 所有处于激活状态的 socket 端口信息。

又如，如果我们要查看本机的路由表，可以使用以下命令：

```
[root@CentOS ~]# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt
Iface
192.168.1.0     0.0.0.0         255.255.255.0    U        0 0          0 eth0
172.16.71.0     0.0.0.0         255.255.255.0    U        0 0          0 eth0
```

4. traceroute 命令

traceroute 命令用来显示网络数据传输的路径信息，以便追踪数据传输的路由状态，从而查找网络问题。其使用格式为：

```
traceroute [选项] [IP 地址或者主机名] [数据包大小]
```

traceroute 命令的可用选项及其具体含义如表 3-46 所示。

表 3-46　traceroute 命令选项及其含义

选　项	含　义
-I 网络接口	使用指定的网络接口收发数据
-n	使用 IP 地址而不是主机名
-v	详细显示命令的执行过程
-w 秒数	设置等待远程主机回应的最大时间
-x	开启或者关闭效验数据包中的数据
-s IP 地址	设置本地主机发送数据的 IP 地址
-g 网关 IP	设置来源的路由网关

例如，要查看从本机到访问 www.baidu.com 的数据走向，可以使用以下命令：

```
[root@CentOS ~]# traceroute -i eth0 -w 10 www.baidu.com 100
traceroute to www.baidu.com (119.75.218.70), 30 hops max, 100 byte
```

```
packets
 1  192.168.111.2 (192.168.111.2)  0.161 ms  0.166 ms  0.118 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
…
```

从上面的例子我们可以看出 traceroute 的最大测试跳数为 30 跳，但由于网络存在防火墙，所以一些数据包不能被正确地反馈。

5. telnet 命令

telnet 命令的功能是使用 telnet 协议与远程主机建立连接。其使用格式为：

```
telnet [主机名或者 IP 地址] [端口]
```

6. wget 命令

wget 命令用来从网络中下载文件，它是在 Linux 中常用的网络下载命令，其使用格式为：

```
wget [下载文件网址]
```

例如，我们已知 Linux3.0.1 内核的下载地址，那么可以使用 wget 命令将其下载到本机中：

```
[root@CentOS /]# wget \
> http://www.kernel.org/pub/linux/kernel/v3.x/linux-3.0.1.tar.gz
--2012-02-17 11:41:52--
http://www.kernel.org/pub/linux/kernel/v3.x/linux-3.0.1.tar.gz
Resolving www.kernel.org... 149.20.4.69
Connecting to www.kernel.org|149.20.4.69|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 96687674 (92M) [application/x-gzip]
Saving to: `linux-3.0.1.tar.gz'

 0% [                                ] 437,452    23.2K/s  eta 55m 28s
```

3.2.6 使用 vi 文本编辑工具

vi 是 Linux 提示符界面最常用的文本编辑工具，它允许用户查看、搜索和修改文本文件。几乎所有的 Linux 版本都加入了对 vi 工具的支持，所以，读者熟练掌握 vi 工具的使用也是非常必要的。

1. 启动 vi

要启动 vi 工具，可以在 Linux 提示符下直接输入“vi”命令，即可打开一个新的文本文件，如图 3-2 所示。还会显示 vi 编辑器的版本及帮助信息。

或者，如果要使用 vi 编辑器打开某个已经存在的文本文件，可以在提示符中输入如下的命令格式：

```
vi [选项][文件名]
```

```
VIM - Vi IMproved

version 7.0.237
by Bram Moolenaar et al.
Vim is open source and freely distributable

Become a registered Vim user!
type  :help register<Enter>   for information

type  :q<Enter>               to exit
type  :help<Enter>  or  <F1>  for on-line help
type  :help version7<Enter>   for version info
```

图 3-2　打开 vi 文本编辑器

vi 命令可用的选项及其含义如表 3-47 所示。

表 3-47　vi 命令选项及其含义

选　项	含　义
-r	恢复上一次 vi 打开时崩溃的文件
-R	以只读方式打开文件
+	打开文件并将光标置于最后一行
+n	打开文件并将光标置于第 n 行
+pattern	打开文件，并将光标置于第一个与 pattern 匹配的位置
+c 命令	打开文件之前先执行指定的命令

2. vi 的操作模式

vi 编辑器有两种基本的操作模式：命令模式和输入模式。在默认配置下，vi 将以命令模式打开文件。

- 所谓命令模式，是指在该模式下输入的按键将作为操作指令来处理，例如输入 a，即认为是要在当前位置插入字符。
- 所谓输入模式，是指在该模式下可以在文档中修改和输入字符。

从命令模式切换到输入模式，只需要输入相应的命令即可(我们将稍后介绍这些命令)，而从输入模式切换到命令模式，只需按 Esc 键。

3. vi 编辑器的环境变量

vi 的环境变量有很多，可以使用“:set all”命令来查看所有 vi 可用的环境变量，如图 3-3 所示。

使用环境变量的格式为：

```
:set [变量名及参数]
```

```
:set all
--- Options ---
  ambiwidth=single  nohidden            nopreserveindent      termencoding=
noautoindent          history=50          prompt            noterse
noautoread          nohlsearch          noreadonly              textauto
noautowrite         noignorecase          remap             notextmode
noautowriteall        iminsert=0          report=2              textwidth=0
  background=light    imsearch=0          scroll=12         notildeop
nobackup            noincsearch           scrolljump=1          timeout
  backupcopy=auto   noinfercase           scrolloff=0           timeoutlen=1000
  backupext=~       noinsertmode        nosecure            nottimeout
  backupskip=/tmp/*   isprint=@,161-255   shell=/bin/bash       ttimeoutlen=-1
nobinary              joinspaces          shellcmdflag=-c       ttybuiltin
nobomb                keywordprg=man      shellquote=           ttyfast
  buflisted         nolazyredraw          shelltemp             ttyscroll=999
  cmdheight=1         lines=24            shellxquote=          ttytype=xterm
  columns=80        nolist              noshiftround            undolevels=1000
nocompatible          listchars=eol:$     shiftwidth=8          updatecount=200
nocopyindent          loadplugins       noshortname             updatetime=4000
  cpoptions=aABceFs   magic             noshowfulltag           verbose=0
  debug=              matchtime=5       noshowmatch             verbosefile=
nodelcombine          maxcombine=2        showmode          novisualbell
  display=            maxmapdepth=1000    sidescroll=0          warn
-- More --
```

图 3-3　vi 环境变量

vi 常用的环境变量如表 3-48 所示。

表 3-48　vi 常用环境变量及其含义

环境变量	含　义
number	显示每行的行号
readonly	以只读方式打开文件
autowrite	使文件在[:n]和[:!]命令之前都自动保存
showmode	显示用户当前处于什么模式下
noshowmode	不显示用户当前处于什么模式下

4. vi 的基本操作

vi 编辑器在命令模式下的指令，主要包括移动光标、屏幕滚动、插入和修改文本、搜索和替换以及保存退出等几类命令，具体的命令如表 3-49 所示。

表 3-49　vi 命令模式指令及其含义

指　令	含　义
h	光标左移一个字符
l	光标右移一个字符
Backspace 键	光标左移一个字符
k 或者 Ctrl+p	光标上移一个字符
j 或者 Ctrl+n	光标下移一个字符
Enter 键	光标下移一行
w 或者 b	光标右移到字首
e	光标右移到字尾
nG	光标移动到第 n 行
n+	光标下移 n 行

续表

指　令	含　义
n-	光标上移 n 行
n$	相对于现在所在行，光标再后移 n 行
H	光标移动到当前屏幕顶行
M	光标移动到当前屏幕中间行
L	光标移动到当前屏幕的最后一行
0	光标移动到当前行首
$	光标移动到当前行尾
:$	光标移动到文件最后一行的行首
Ctrl+u	向前滚动半屏
Ctrl+d	向后滚动半屏
Ctrl+b	向前滚动一屏
Ctrl+f	向后滚动一屏
nz+Enter	将文件的第 n 行滚动到屏幕顶部，如果不指定 n 的值，则是将光标当前所在行滚动到屏幕顶部
ESC	返回命令模式
I	在光标前插入文字
a	在光标后插入文字
o	在当前行之下插入文字
O	在当前行之上插入文字
r	替换光标当前所在的字符
x	删除光标所在的字符
dd	删除光标所在的行，删除后的内容自动保存在剪贴板中
nyy	将光标所在的行及其下面的 n-1 行复制到剪贴板，如果不指定 n 的值，则只复制一行
p	将剪贴板中的内容复制到光标后
P	将剪贴板中的内容复制到光标前
yw	将光标所在的单词复制到剪贴板
/abc	在文件中向前查找字符串 abc
?abc	在文件中向后查找字符串 abc
n	在同一方向重复上次的搜索操作
N	在反方向重复上次的搜索操作
:s/a1/a2/g	将当前光标所在行中的所有 a1 替换为 a2
:n1,n2ss/a1/a2/g	将文件中第 n1 行至 n2 行中的所有 a1 替换为 a2
:g/a1/a2/g	将文件中所有的 a1 替换为 a2
:wq	保存并退出 vi 编辑器
:wq!	不保存文档，直接退出 vi 编辑器

续表

指　令	含　义
:q	不保存文档，退出 vi 编辑器
:w	保存文档
x!	保存文件，退出 vi 编辑器

3.3　Linux 常用网络配置文件

在本章前面的讲述中我们已经提到，使用网络配置命令来配置网卡的地址信息有一个很大的缺点就是当网卡或者主机重启后所有配置信息会丢失，这一点在服务器应用中是非常不利的，为了能够长期使用静态 IP 地址，我们需要对 Linux 的网络配置文件进行设置。

3.3.1　网络配置文件的位置

不同发行版本的 Linux，其网络配置文件的位置略有不同，下面以 CentOS 为例，介绍配置网络文件的路径。

网络配置文件都在/etc 目录中，具体路径为：

```
/etc/sysconfig/network-scripts/ifcfg-(网卡名)    #根据网卡类型的不同，网卡配置
文件的命名也会有所不同，一般以 eth0 或者 ppp0 等命名。
/etc/sysconfig/network-scripts/ifcfg-lo    #回环网卡
/etc/sysconfig/network                     #主机名和网关配置文件
/etc/tesolv.conf                           #DNS 配置文件
/etc/hosts                                 #设置主机和 IP 地址绑定文件
```

3.3.2　网络配置文件解析

修改网络配置文件主要包括对网卡文件、回环配置文件、主机名和网关配置文件、DNS 配置文件以及主机 IP 地址绑定文件的配置。

1) 网卡配置文件

打开网卡配置文件/etc/sysconfig/network-scripts/ifcfg-eth0，显示内容如下：

```
[root@CentOS network-scripts]# more ifcfg-eth0
DEVICE=eth0                 #网卡名称
ONBOOT=yes                  #开机启动
BOOTPROTO=static            #IP 地址获得方式
BROADCAST=192.168.60.255    #广播地址
IPADDR=192.168.60.2         #IP 地址
NETMASK=255.255.255.0       #子网掩码
NETWORK=192.168.60.0        #网络号
GATEWAY=192.168.60.1        #网关地址
HWADDR=00:0c:29:02:50:58    #MAC 地址
TYPE=Ethernet               #网络类型
```

其中：

- "DEVICE"字段给出了网卡设备的名称，默认系统会根据网卡的类型给出相应的命名，用户也可以自己定义。
- "ONBOOT"字段设置网卡是否开机自动启动，如果用户不希望开机自动运行，可以设置为"no"。
- "BOOTPROTO"字段未设置网卡地址的获得方式，此处为"静态(Static)"，如果用户希望通过 DHCP 服务器分配，可以设置为"DHCP"。
- "BROADCAST、IPADDR、NETMASK、NETWORK、GATEWAY"分别给出了广播地址、IP 地址、子网掩码、网络号、网关地址的信息。在静态获得 IP 地址时，系统会根据这些配置信息来配置网卡。
- "HWADDR"字段给出了网卡的物理地址，一般情况下由系统自动获得。
- "TYPE"字段则给出了网卡所在网络的类型，此处为"以太网(Ethernet)"。

知道了这些字段的含义后，用户可以根据需要为主机设置网卡的配置信息，配置完成后，使用下面的命令使网络设置生效：

```
service network restart
```

如果用户希望在一个网络设备中绑定多个 IP 地址，只需要在/etc/sysconfig/network-scripts/目录中在创建一个以新的名为文件名的文件，例如"ifcfg-rth0:0"，然后将原网卡配置文件的信息复制到新的文件中，修改其"DEVICE"值为 eth0:0，修改新的 IP 地址信息即可，例如下面的例子是 eth0:0 的 IP 配置信息。

```
DEVICE=eth0:0                 #网卡名称
ONBOOT=yes                    #开机启动
BOOTPROTO=static              #IP 地址获得方式
BROADCAST=192.168.60.255      #广播地址
IPADDR=192.168.60.3           #IP 地址
NETMASK=255.255.255.0         #子网掩码
NETWORK=192.168.60.0          #网络号
GATEWAY=192.168.60.1          #网关地址
```

2) 配置回环配置文件

打开回环配置文件/etc/sysconfig/network-scripts/ifcfg-lo，显示内容如下：

```
[root@CentOS ~]# more /etc/sysconfig/network-scripts/ifcfg-lo
DEVICE=lo
IPADDR=127.0.0.1
NETMASK=255.0.0.0
NETWORK=127.0.0.0
# If you're having problems with gated making 127.0.0.0/8 a martian,
# you can change this to something else (255.255.255.255, for example)
BROADCAST=127.255.255.255
ONBOOT=yes
NAME=loopback
```

其中，"DEVICE"为系统默认设置，"NAME"字段的值 loopback 指出了这是一个回环地址。

3) 主机名和网关配置文件

打开文件/etc/sysconfig/network，显示内容如下所示：

```
[root@CentOS ~]# more /etc/sysconfig/network
NETWORKING=yes
NETWORKING_IPV6=yes
HOSTNAME=CentOS
GATEWAY=192.168.60.1
```

其中：

- “NETWORKING”字段指出当前网络是否运行正常，为系统自动设置。
- “NETWORKING_IPV6”表示是否支持 IPv6 网络，如果用户不希望开启 IPv6 网络，可以设置为“no”。
- “HOSTNAME”则设置了主机名。
- “GATEWAY”的含义与 ifcfg-eth0 中的相同，都是设置网关 IP 地址。

4) 配置 DNS 文件

打开文件/etc/resolv.conf，显示内容如下：

```
[root@CentOS ~]# more /etc/resolv.conf
; generated by /sbin/dhclient-script
nameserver 192.168.44.2
```

其中：“nameserver”即表示域名服务器，其后的 IP 地址为 DNS 服务器。

5) 设置主机和 IP 绑定

打开/etc/hosts 文件，显示内容如下：

```
[root@CentOS ~]# more /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       CentOS localhost.localdomain localhost
::1            localhost6.localdomain6 localhost6
```

hosts 文件用来实现本地解析功能，其中每行代表了一个本地解析的域名及地址信息，每行包含三条信息：

- 网络 IP 地址。
- 主机名.域名。
- 主机名。

3.4 本 章 小 结

本章主要介绍了 Linux 中常用的配置命令及其使用方法，以及在 Linux 中常用的配置文件的位置及配置方法。Linux 系统命令非常丰富，需要读者在实践中不断地积累经验来加深记忆，以达到快速上手的目的。

3.5 课 后 习 题

1．填空题

(1) 在 Linux 中，若要为命令“ls -art”设置一个别名 tdir，则应在命令行中输入别名

命令：________________________。从命令行设置的别名只在当前会话中有效。为想在登录时使别名有效，如果你使用的是 bash，则把这个别名定义放在用户主目录中的_________文件或____________文件中。

(2) 在 Linux 中，用户可通过____命令来创建文件链接。链接有两种，一种被称为_______(这类链接也通常被称为一般链接)，它要求链接文件和被链接文件必须位于同一个文件系统中，并且不能链接目录。另一种被称为____________的链接方式则不存在这一问题。

(3) 在 Linux 中，用户可通过 cat 命令来创建一个新文件。若要创建新文件 abc，则应在命令行中输入____________命令。然后，用户可通过键盘输入文件内容，输入完后按 Enter 键，然后按_________组合键或_________组合键来结束输入过程即可。另外，用户还可以通过 cp 命令来创建一个新文件。若一个位于第一个虚拟终端号上的用户要通过 cp 命令创建新文件 abc，则你需在命令行上输入_________________________命令。

2. 选择题

(1) 下列(　　)指令可以用来改变 shell 的使用种类。

A. chown　　B. chmod　　C. chsh　　D. chpwd

(2) bash 是指一种(　　)。

A. shell　　B. batch command　　C. cgi　　D. asp

(3) cd .. 这个指令的用途是(　　)。

A. 切换到自家目录　　B. 离线

C. 回到先前的路径位置　　D. 回到上一层目录

(4) 下列(　　)指令可以用来查看 CPU 的信息。

A. cpu　　B. mem　　C. uname　　D. cat /proc/cpuinfo

3. 判断题

tar 命令只能进行打包或解包操作，没有压缩功能，用户要进行压缩操作，必须使用其他诸如 gzip 之类的压缩软件。　　(　　)

4. 简答题

(1) 简述 shell 命令解释语言的执行过程。

(2) 简述每个网络配置文件的主要功能。

第 4 章

DHCP 服务器安装与配置

在 TCP/IP 网络上，每台计算机在使用网络之前，都要进行基本的网络配置，一些主要的参数，如 IP 地址、子网掩码、默认网关、DNS 等都是必不可少的。在配置这些参数时有两种方法：一种是静态手工配置；另一种是使用 DHCP 服务器动态配置。

在规模较大的网络中，确保所有主机都拥有正确的网络配置是一项相当困难的任务，尤其对于含有漫游用户和笔记本电脑的动态网络更是如此。如果经常有计算机从一个子网移动到另一个子网或从网络中移出，手动配置或重新配置数量巨大的计算机可能要花费很长的时间，而采用手工配置 IP 过程中的错误可能导致该主机无法与网络中其他主机通信。因此我们需要一种机制来简化 IP 地址的配置，实现 IP 地址的集中式管理。IETF(Internet 网络工程师任务小组)设计的动态主机配置协议 DHCP 正好可以解决这个问题。

DHCP 用于为计算机自动提供 IP 地址、子网掩码和路由等网络配置信息。本章从 DHCP 服务器的原理、安装、配置等方面介绍 DHCP 服务。

4.1 DHCP 服务概述

4.1.1 DHCP 简介

动态主机配置协议 DHCP 是 Dynamic Host Configuration Protocol 的缩写。DHCP 的前身是 BOOTP，它工作在 OSI 的应用层，是一种帮助计算机从指定的 DHCP 服务器获取配置信息的自举协议。DHCP 的守护进程为 dhcpd。守护进程使用的端口号为 UDP 67 端口。

DHCP 使用客户端/服务器(Client/Server)的模式，请求配置信息的计算机叫做“DHCP 客户端”，而提供信息的叫做“DHCP 服务器”。DHCP 为客户端分配地址的方法有 3 种，即手工配置、自动配置和动态配置。DHCP 最重要的功能就是动态分配，除了 IP 地址，DHCP 还为客户端提供其他的配置信息，如子网掩码、默认网关、DNS 等信息，从而使得客户端无须用户动手即可完成自动配置并连接网络。

DHCP 服务器将 TCP/IP 网络设置集中起来，动态处理客户端 IP 地址的配置，当设备接入到网络中时，它们会向 DHCP 服务器请求一个 IP 地址。DHCP 服务器为每个请求的设备分配一个 IP 地址，直到分配完该范围内所有 IP 地址为止。已经分配的 IP 地址必须定时延长租期，这个延期的过程叫做 leasing，确保了客户机设备在正常释放 IP 地址之前突然断网时服务器可将其 IP 地址收回。用 DHCP 租约和预置 IP 地址相联系，使 DHCP 服务器实现了在 TCP/IP 网络上安全地分配和租用 IP 地址的机制，完成了 IP 地址的集中式管理，而不再需要管理员的人为干预。

4.1.2 DHCP 的优点

DHCP 简化了客户机的网络参数配置和管理工作。对于 IP 地址与 TCP/IP 相关参数的分配，基本上不需要网络管理员的人为干预。网络中的 DHCP 服务器自动的为 DHCP 客户端分配 IP 地址及 TCP/IP 的配置信息。

DHCP 在快速发送客户端网络配置方面很有用，当配置客户端系统时，若管理员选择 DHCP，则不必输入 IP 地址、子网掩码、网关或 DNS 服务器，客户端从 DHCP 服务器中检索这些信息。DHCP 在网络管理员想改变网络的 IP 地址时也有用，与其重新配置所有系统，不如编辑服务器中的一个用于新 IP 地址集合的 DHCP 配置文件。如果某机构的 DNS 服务器改变，这种改变只需在 DHCP 服务器中，而不必在 DHCP 客户端上进行。一旦客户端的网络被重新启动(或客户端重新引导系统)，改变就会生效。除此之外，如果便携电脑或任何类型的可移动计算机被配置使用 DHCP，只要每个办公室都有一个允许其联网的 DHCP 服务器，它就可以不必重新配置而在办公室间自由移动。

在网络中应用 DHCP 有以下优点：

- 减少错误。通过配置 DHCP，把手工配置 IP 地址所导致的错误减少到最低程度，例如已分配的 IP 再次分配给另一设备所造成的地址冲突等将大大减少。
- 减少网络管理。TCP/IP 配置是集中化和自动完成的，不需要网络管理员手工配

置。网络管理员能集中定义全局和特定子网的 TCP/IP 配置信息。使用 DHCP 选项可以自动给客户机分配全部范围的附加 TCP/IP 配置值。

当然 DHCP 也有一些缺点：DHCP 不能发现网络上非 DHCP 客户机已经在使用的 IP 地址；当网络上存在多个 DHCP 服务器时，一个 DHCP 服务器不能查出已被其他服务器租出去的 IP 地址；DHCP 服务器不能跨路由器与客户机通信，除非路由器允许 BOOTP 转发。

4.1.3　DHCP 的工作流程

DHCP 是一个基于广播的协议。客户机通过广播向服务器申请 IP 地址，服务器将 IP 地址并将其他 TCP/IP 网络配置信息发送给客户机。申请的过程可归纳为 4 个阶段：发现阶段、提供阶段、选择阶段、确认阶段，如图 4-1 所示。



图 4-1　DHCP 工作流程

1．发现阶段

即 DHCP 客户端查找 DHCP 服务器的阶段。客户机以广播方式(因为 DHCP 服务器的 IP 地址对于客户端来说是未知的)发送 DHCPDISCOVER 信息来查找 DHCP 服务器，即向地址 255.255.255.255 发送特定的广播信息，目的端口号为 67。网络上每一台安装了 TCP/IP 的主机都会接收到这种广播信息，但只有 DHCP 服务器才会做出响应。

2．提供阶段

即 DHCP 服务器提供 IP 地址的阶段，在网络中接收到 DHCPDISCOVER 信息的 DHCP 服务器都会做出响应。它从尚未出租的 IP 地址中挑选一个分配给 DHCP 客户端，向其发送一个包含出租的 IP 地址、子网掩码、DHCP 服务器地址、IP 租用期限等数据的 DHCPOFFER 数据包。由于客户端在开始的时候没有 IP 地址，所以 DHCPDISCOVER 包内会包含客户端 MAC 地址信息并有一个 xid 编号用来标识该包。DHCP 服务器会根据这些信息将 DHCPOFFER 包发送给客户端。

注意： MAC 地址也叫物理地址、硬件地址或链路地址，由 6 字节 48bit 组成。该地址是由 OSI 参考模型中数据链路层中的媒体访问控制子层定义，由网络设备制造商生产时写在网络硬件内部。一般来说，MAC 地址是全球唯一的。

3．选择阶段

即 DHCP 客户端选择某台 DHCP 服务器提供的 IP 地址的阶段。如果有多台 DHCP 服务器向 DHCP 客户端发送 DHCPOFFER 信息，则 DHCP 客户端会在其中挑选一个

DHCPOFFER 信息(通常是最先到达的那个)。然后它就以广播方式回答一个 DHCPREQUEST 信息，该信息中包含向它所选定的 DHCP 服务器请求 IP 地址的内容。之所以要以广播方式回答，是为了通知所有 DHCP 服务器，它将选择某台 DHCP 服务器所提供的 IP 地址。

同时 DHCP 客户端还会向网络中发送一个 arp 包，如果查询到网络中该 IP 地址是否已经被使用，客户端将发送一个 DHCPDECLINE 包给 DHCP 服务器，并重新发送 DHCPDISCOVER 信息。

4．确认阶段

即 DHCP 服务器确认所提供的 IP 地址的阶段。当 DHCP 服务器收到 DHCP 客户端回答的 DHCPREQUEST 信息之后，它向 DHCP 客户端发送一个包含其所提供的 IP 地址和其他设置的 dhcpack 信息，告诉 DHCP 客户端可以使用该 IP 地址，然后 DHCP 客户端便将其 TCP/IP 与网卡绑定。另外，除 DHCP 客户端选中的服务器外，其他的 DHCP 服务器都将收回曾提供的 IP 地址。

对于那些已经分配 IP 且租约尚未到期的客户端，客户端每次重新登录网络时，不需要发送 DHCPDISCOVER 信息，而是直接发送包含前一次所分配的 IP 地址的 DHCPREQUEST 信息。当 DHCP 服务器收到这一信息后，它会尝试让 DHCP 客户端继续使用原来的 IP 地址，并回答一个 dhcpack 信息。如果此 IP 地址已无法再分配给原来的 DHCP 客户端使用(比如此 IP 地址已分配给其他 DHCP 客户端使用)，则 DHCP 服务器给 DHCP 客户端回答一个 dhcpack 信息。当原来的 DHCP 客户端收到此信息后，必须重新发送 DHCPDISCOVER 信息来请求新的 IP 地址。

下面，介绍一下 DHCP 更新租约的过程，如图 4-2 所示。DHCP 服务器向 DHCP 客户端出租的 IP 地址一般都有一个租借期限，期满后 DHCP 服务器便会收回该 IP 地址。如果 DHCP 客户端要延长其 IP 租约，则必须更新其 IP 租约。DHCP 客户端启动时和 IP 租约期限过一半时，DHCP 客户端都会自动向 DHCP 服务器发送更新其 IP 租约的信息。

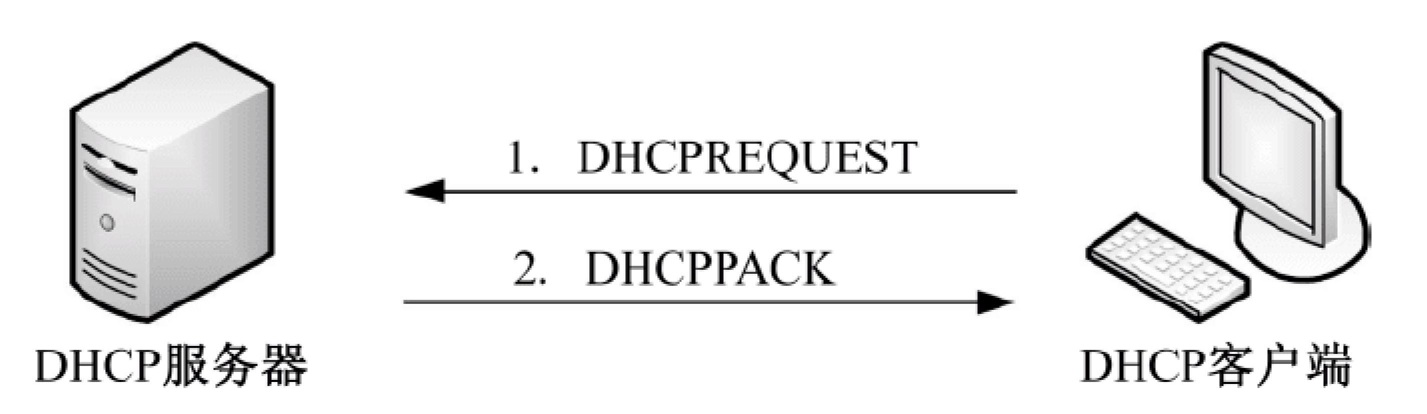


图 4-2 DHCP 更新租约

注意： 由于 DHCP 是基于广播的协议，因此在一般情况下服务器和客户机应位于同一个网络中。对于跨网段的 DHCP 可以使用 DHCP 中继来实现，后面章节将做详细介绍。

4.1.4 DHCP 术语

DHCP 术语如表 4-1 所示。

表 4-1　DHCP 术语

术　语	描　述
作用域	“作用域”是网络上可能的 IP 地址的完整连续范围。作用域通常定义为接受 DHCP 服务的网络上的单个物理子网。作用域还为网络上的客户端提供服务器对 IP 地址及任何相关配置参数的分发和指派进行管理的主要方法
超级作用域	“超级作用域”是作用域的管理组合，它可用于支持同一物理子网上的多个逻辑 IP 子网。超级作用域仅包含可同时激活的“成员作用域”或“子作用域”列表。超级作用域不用于配置有关作用域使用的其他详细信息。如果想配置超级作用域内使用的多数属性，需要单独配置成员作用域属性
排除范围	“排除范围”是作用域内从 DHCP 服务中排除的有限 IP 地址序列。排除范围确保服务器不会将这些范围中的任何地址提供给网络上的 DHCP 客户端
地址池	在定义了 DHCP 作用域并应用排除范围之后，剩余的地址在作用域内形成可用的“地址池”。服务器可将池内地址动态地指派给网络上的 DHCP 客户端
租约	“租约”是由 DHCP 服务器指定的一段时间，在此时间内客户端计算机可使用指派的 IP 地址。当向客户端提供租约时，租约是“活动”的。在租约过期之前，客户端通常需要向服务器更新指派给它的地址租约。当租约过期或在服务器上被删除时，它将变成“非活动”的。租约期限决定租约何时期满以及客户端需要向服务器对它进行更新的频率
保留	可使用“保留”创建 DHCP 服务器指派的永久地址租约。保留可确保子网上指定的硬件设备始终可使用相同的 IP 地址
选项类型	“选项类型”是 DHCP 服务器在向 DHCP 客户端提供租约时可指派的其他客户端配置参数。例如，一些常用选项包含用于默认网关(路由器)、DNS 服务器的 IP 地址。通常，为每个作用域启用并配置这些选项类型。DHCP 控制台还允许您配置由服务器上添加和配置的所有作用域使用的默认选项类型。虽然大多数选项都是在 RFC 2132 中预定义的，但若需要时也可使用 DHCP 控制台定义并添加自定义选项类型

4.2　DHCP 服务的安装与运行

4.2.1　安装 DHCP 服务器

CentOS 5 在默认安装时不会安装 DHCP 服务器软件包，用户可以在安装系统后单独安装。安装步骤如下：

(1) 检查 Linux 系统是否安装了 DHCP 服务软件包。

```
# rpm -qa | grep dhcp
```

(2) 加载 CentOS 光盘，进入光盘目录的 CentOS 文件夹。

```
#cd /media/CentOS_5.*/CentOS
```

(3) 安装 DHCP 服务器软件包。

```
# rpm -ivh dhcp-*
```

显示结果如下：

```
Preparing...          ########################################### [100%]
   1:dhcp          ########################################### [ 50%]
   2:dhcp-devel    ########################################### [100%]
```

(4) 再次检查 DHCP 服务器软件包安装情况。

```
# rpm -qa | grep dhcp
```

显示结果如下所示，即说明 DHCP 服务器已经安装成功。

```
dhcp-3.0.5-23.el5
dhcpv6-client-1.0.10-20.el5
dhcp-devel-3.0.5-23.el5
```

dhcp-3.0.5-23.el5 软件包为 DHCP 服务器软件包，是必须安装的。dhcpv6-client-1.0.10-20.el5 软件包为 IPv6 协议下支持动态配置 IPv6 地址的 DHCP 客户端；dhcp-devel-3.0.5-23.el5 为 DHCP 的 API 开发包，包含 DHCP 中的所有的类库和头文件。在光盘软件包中还有一个 dhcpv6-1.0.10-18.el5.i386.rpm，是用于 IPv6 下 DHCP 服务器软件的。

4.2.2 启动 DHCP 服务器

1. 启动 DHCP 服务器

可以使用以下两种命令启动 DHCP 服务器：

```
# service dhcpd start
启动 dhcpd：                                               [确定]

# /etc/init.d/dhcpd start
启动 dhcpd：                                               [确定]
```

2. 停止 DHCP 服务器

使用以下两种命令可以停止 DHCP 服务器的运行：

```
# service dhcpd stop
关闭 dhcpd：                                               [确定]

# /etc/init.d/dhcpd stop
关闭 dhcpd：                                               [确定]
```

3. 重新启动 DHCP 服务器

使用以下两种命令可以重新启动 DHCP 服务器：

```
# service dhcpd restart
关闭 dhcpd：                                               [确定]
启动 dhcpd：                                               [确定]

# /etc/init.d/dhcpd restart
关闭 dhcpd：                                               [确定]
启动 dhcpd：                                               [确定]
```

4. 使用 ps 命令检查 dhcpd 进程情况

```
# ps -ef | grep dhcpd
root     21859     1  0 19:36 ?        00:00:00 /usr/sbin/dhcpd
root     29110 21833  0 22:59 pts/1    00:00:00 grep dhcpd
```

5. 使用 netstat 检查 dhcpd 运行的端口

```
# netstat -nutap | grep dhcpd
udp        0      0 0.0.0.0:67              0.0.0.0:*     21859/dhcpd
```

6. 设置 DHCP 服务器开机自启动

```
# chkconfig --level 345 dhcpd  on
```

7. 使用图形化方式设置 DHCP 服务器

CentOS 5 中自带了一种很好的设置服务器各种服务的工具。选择“系统”→“管理”→“服务器设置”→“服务”，打开 CentOS 中的“服务配置”窗口，如图 4-3 所示。在该窗口中选中 dhcpd 服务，然后单击“开始”、“停止”、“重启”等按钮即可实现对 dhcp 服务的启动、停止和重启操作。在窗口中选中 dhcpd 复选框，还可以实现 dhcpd 服务的开机自动运行。

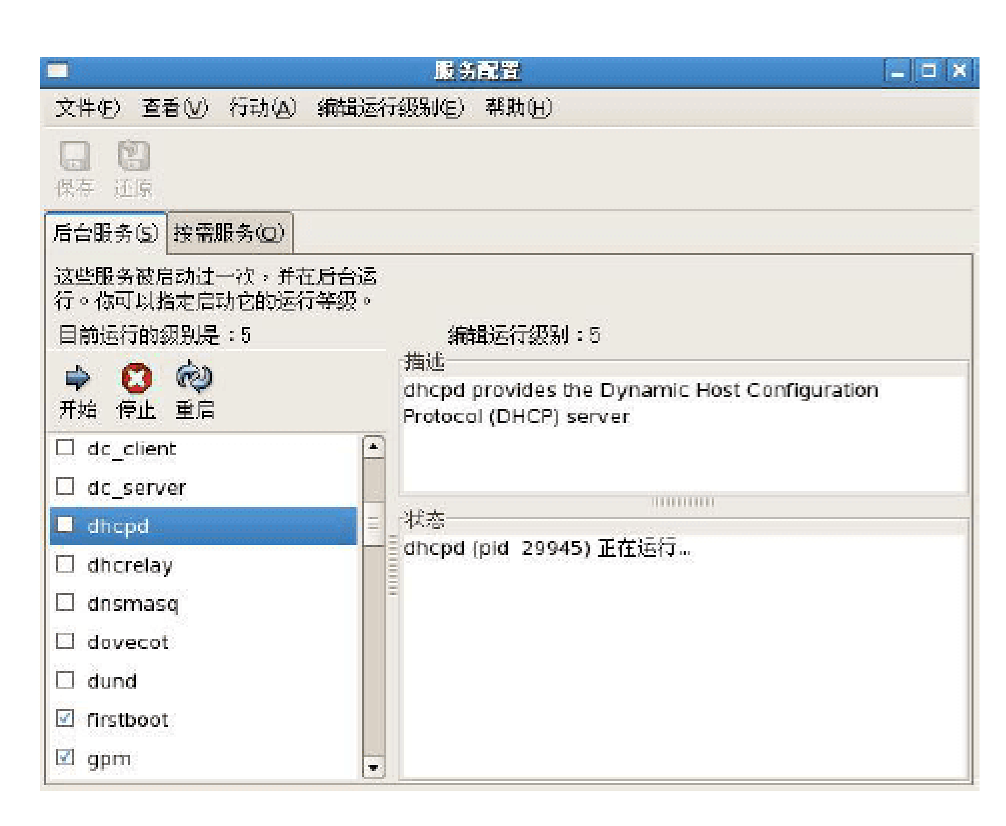


图 4-3　CentOS 5 中的服务配置

4.3　DHCP 服务的配置文件

DHCP 服务的主要配置文件有两个：一个是主配置文件/etc/dhcpd.conf；另一个是网卡启动文件/etc/sysconfig/dhcpd。下面将对这些文件进行介绍。

注意： 不同版本的 CentOS 所配置的 DHCP 软件版本不同，其实际的安装路径也不尽相同。如 CentOS 6.2 默认包含的 DHCP 版本为 4.1.1，其配置文件所在目录为/etc/dhcp/dhcpd.conf。我们可以使用 rpm -ql dhcp 命令来查看 dhcp 软件包中每个文件的具体位置。

4.3.1 DHCP 主配置文件

DHCP 主配置文件是/etc/dhcpd.conf 文件。在 CentOS 5 中，DHCP 服务软件包安装完成后主配置文件/etc/dhcpd.conf 没有包含配置信息，需要用户从 DHCP 服务器模板中复制或者手工创建主配置文件。例如，在 CentOS 5.5 中安装 DHCP 服务后可查看主配置文件内容：

```
# more /etc/dhcpd.conf
```

显示内容如下所示：

```
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.sample
#
```

用户可以直接将 DHCP 的配置模板/usr/share/doc/dhcp*/dhcpd.conf.sample 中的内容替换为/etc/dhcpd.conf 即可：

```
#cp /usr/share/doc/dhcp*/dhcpd.conf.sample  /etc/dhcpd.conf -f
```

查看此配置文件：

```
# more /etc/dhcpd.conf
```

可得到配置文件内容如下：

```
ddns-update-style interim;
ignore client-updates;
subnet 192.168.0.0 netmask 255.255.255.0 {
# --- default gateway
       option routers                  192.168.0.1;
       option subnet-mask               255.255.255.0;
       option nis-domain               "domain.org";
       option domain-name              "domain.org";
       option domain-name-servers        192.168.1.1;
       option time-offset              -18000; # Eastern Standard Time
#      option ntp-servers               192.168.1.1;
#      option netbios-name-servers       192.168.1.1;
# --- Selects point-to-point node (default is hybrid). Don't change this
unless
# -- you understand Netbios very well
#      option netbios-node-type 2;
       range dynamic-bootp 192.168.0.128 192.168.0.254;
       default-lease-time 21600;
       max-lease-time 43200;
 # we want the nameserver to appear at a fixed address
       host ns {
              next-server marvin.redhat.com;
              hardware ethernet 12:34:56:78:AB:CD;
              fixed-address 207.175.42.254;
       }
}
```

DHCP 服务主配置文件的基本规范如下：

- “#”为注释符号。

- 除在右括号“)”后面外，其他每一行都要以“；”为结尾。
- 设定项目语法主要为『 <参数代号> <设定内容> 』，例如：default-lease-time 259200。
- 某些设定项目必须以 option 来设定，基本方式为『 option <参数代码> <设定内容> 』，例如：option domain-name "your.domain.name"。

为方便记忆和理解，一般将 DHCP 主配置文件分为 3 个部分：parameters 参数、declarations 声明和 option 选项。

1．DHCP 配置文件中的 parameters 参数

parameters 表明如何执行任务，是否要执行任务或将哪些网络配置选项发送给客户端，主要参数如表 4-2 所示。

表 4-2　DHCP 配置文件中的主要参数

参　数	解　释
ddns-update-style	配置 DHCP-DNS 互动更新模式
default-lease-time	指定默认租赁时间的长度，单位是秒
max-lease-time	指定最大租赁时间的长度，单位是秒
hardware	指定网卡接口类型和 MAC 地址
server-name	通知 DHCP 客户端服务器名称
get-lease-hostnames flag	检查客户端使用的 IP 地址
fixed-address ip	分配给客户端一个固定的地址
authritative	拒绝不正确的 IP 地址的要求

2．DHCP 配置文件中的 declarations 声明

declarations 用来描述网络布局及提供客户的 IP 地址等，主要声明如表 4-3 所示。

表 4-3　DHCP 配置文件中的主要声明

声　明	解　释
shared-network	用来告知是否一些子网络共享相同网络
subnet	描述一个 IP 地址是否属于该子网
range 起始 IP 终止 IP	提供动态分配 IP 的范围
host 主机名称	参考特别的主机
group	为一组参数提供声明
allow unknown-clients；deny unknown-client	是否动态分配 IP 给未知的使用者
allow bootp;deny bootp	是否响应激活查询
allow booting；deny booting	是否响应使用者查询
filename	开始启动文件的名称，应用于无盘工作站
next-server	设置服务器从引导文件中装入主机名，应用于无盘工作站

3．DHCP 配置文件中的 option 选项

option 用来配置 DHCP 可选参数，全部用 option 关键字作为开始，主要选项如表 4-4 所示。

表 4-4　DHCP 配置文件中 option 关键字的主要选项

选　项	解　释
subnet-mask	为客户端设定子网掩码
domain-name	为客户端指明 DNS 名字
domain-name-servers	为客户端指明 DNS 服务器的 IP 地址
host-name	为客户端指定主机名称
routers	为客户端设定默认网关
broadcast-address	为客户端设定广播地址
ntp-server	为客户端设定网络时间服务器的 IP 地址
time-offset	为客户端设定格林尼治时间的偏移时间，单位是秒

在修改主配置文件时，需要注意地址池的范围一定要在子网的有效 IP 地址范围之内。如地址池的 IP 地址范围超出了子网的有效 IP 地址范围，启动或重启 DHCP 服务，系统将提示错误信息。同时本机网卡的 IP 地址也需要在 DHCP 定义的网络范围之内。若用户 IP 不在其网段中，启动或者重启 DHCP 服务时，系统也会提示错误信息。

注意： 如果客户端使用 Windows 操作系统，不要为其指定 host-name 主机名称选项。

4.3.2　DHCP 的网卡启动文件

DHCP 的网卡启动文件为/etc/sysconfig/dhcpd，主要用来设置 DHCP 服务从哪块网卡启动。该配置文件内容如下：

```
# Command line options here
DHCPDARGS=
```

其中，DHCPDARGS 用来设置 DHCP 服务器通过哪块网卡启动，这在多网卡的服务器中是非常有必要的。如 DHCP 服务器只有一块网卡，则无须设置该配置文件。例如通过 eth0 的网卡来启动 DHCP 服务，则可修改/etc/sysconfig/dhcpd 文件，做出更改如下：

```
DHCPDARGS=eth0
```

如果有一个带有两块网卡的防火墙机器，这种方法就会大派用场。一块网卡可以被配置成 DHCP 客户端从互联网上检索 IP 地址；另一块网卡可以被用做防火墙之后的内部网络的 DHCP 服务器。仅指定连接到内部网络的网卡使系统更加安全，因为用户无法通过互联网来连接其守护进程。

其他可在/etc/sysconfig/dhcpd 中指定的命令行选项如下：

(1) -p<portnum>：指定 dhcpd 应该监听的 UDP 端口号码，默认值为 67。DHCP 服务

器在比指定的 UDP 端口大一位的端口号上把回应传输给 DHCP 客户端。例如，如果使用默认端口 67，服务器就会在端口 67 上监听请求，然后在端口 68 上回应客户。如果在此处指定了一个端口号，并且使用了 DHCP 转发代理，所指定的 DHCP 转发代理所监听的端口必须是同一端口。

(2) -f：把守护进程作为前台进程运行，在调试时最常用。

(3) -d：把 DCHP 服务器守护进程记录到标准错误描述器中，在调试时最常用。如果未指定，日志将被写入/var/log/messages 中。

(4) -cf<filename>：指定配置文件的位置，默认为/etc/dhcpd.conf。

(5) -lf<filename>：指定租期数据库文件的位置。如果租期数据库文件已存在，在 DHCP 服务器每次启动时使用同一个文件至关重要。建议只在无关紧要的机器上为调试目的才使用该选项，默认为/var/lib/dhcp/dhcpd.leases。

(6) -q：在启动该守护进程时，不要显示整篇版权信息。

4.3.3　DHCP 服务器端租约文件

运行 DHCP 服务还需要一个名为“dhcpd.leases”的文件，它在服务器端保存所有已经分发的 IP 地址。在 CentOS 5 中，该文件位于/var/lib/dhcpd/目录中。dhcpd.leases 的文件格式为：

```
Leases address {statement}
```

一个典型的租约文件内容如下：

```
# All times in this file are in UTC (GMT), not your local timezone.
This is
# not a bug, so please don't ask about it.  There is no portable way to
# store leases in the local timezone, so please don't request this as a
# feature.  If this is inconvenient or confusing to you, we sincerely
# apologize.  Seriously, though - don't ask.
# The format of this file is documented in the dhcpd.leases(5) manual
page.
# This lease file was written by isc-dhcp-V3.0.5-RedHat

lease 192.168.0.252 {                  #DHCP 服务器分配的 IP 地址
  starts 1 2012/03/19 06:26:16;        # lease 开始租约时间
  ends 1 2012/03/19 12:26:16;          # lease 结束租约时间
  binding state active;                # 租约的绑定状态为激活
  next binding state free;             # 下一个租约的绑定状态为自由
  hardware ethernet 00:0c:29:87:8f:26;      # 客户机网卡的 MAC 地址
}
```

注意： lease 开始租约时间和 lease 结束租约时间是格林尼治标准时间(GMT)，不是本地时间。

第 1 次运行 DHCP 服务器时，dhcpd.leases 是一个空文件，也不用手工建立。如果不是通过 RPM 安装 DHCP 服务器，或者 dhcpd 已经安装，那么应该确保该文件存在。也可以手工建立一个空文件：

```
#touch /var/lib/dhcp/dhcpd.leases
```

4.3.4 DHCP 客户端租约文件

DHCP 客户端租约文件为/var/lib/dhcp/dhclient.leases。该文件用于记录 DHCP 客户的租约信息，如租用的 DHCP 服务器的 IP 地址、子网掩码、网关、DNS 信息、域名、最小租期、最大租期等。

客户端租约文件格式如下：

```
lease {statement}
```

相同的网卡接口可能有多条记录，一般以最后的记录为准。/var/lib/dhclinet/dhclient.lease 内容如下：

```
lease {
  interface "eth0";
  fixed-address 192.168.0.252;
  option subnet-mask 255.255.255.0;
  option time-offset -18000;
  option dhcp-lease-time 21600;
  option routers 192.168.0.1;
  option dhcp-message-type 5;
  option dhcp-server-identifier 192.168.0.1;
  option domain-name-servers 192.168.1.1;
  option nis-domain "domain.org";
  option domain-name "domain.org";
  renew 1 2012/3/19 14:15:51;
  rebind 1 2012/3/19 16:52:04;
  expire 1 2012/3/19 17:37:04;
}
lease {                                   #最后的记录为最近生效的记录
  interface "eth0";                       #端口 eth0
  fixed-address 192.168.0.252;            #分配到的 IP 地址
  option subnet-mask 255.255.255.0;       #分配到的子网掩码
  option time-offset -18000;              #时区差
  option routers 192.168.0.1;             #网关地址
  option dhcp-lease-time 21600;           #DHCP 释放时间
  option dhcp-message-type 5;             #DHCP 消息类型
  option domain-name-servers 192.168.1.1;        #DNS 服务器 IP 地址
  option dhcp-server-identifier 192.168.0.1;     #DHCP 服务器 IP 地址
  option nis-domain "domain.org";                #NIS 域名
  option domain-name "domain.org";               #域名
  renew 1 2012/3/19 13:55:51;             #更新时间
  rebind 1 2012/3/19 16:54:34;            #重新绑定时间
  expire 1 2012/3/19 17:39:34;            #过期时间
}
```

4.4 DHCP 服务器的配置

4.4.1 DHCP 服务器配置步骤

一般情况下，DHCP 服务器的配置步骤如下：

(1) 创建 DHCP 服务器主配置文件，可从/usr/share/doc/dhcp*/dhcpd.conf.sample 模板文件中复制。

(2) 修改/etc/dhcpd.conf 主配置文件内容。

(3) 配置 DHCP 网卡启动接口，修改/etc/sysconfig/dhcpd 配置文件。

(4) 根据 DHCP 主配置文件的 subnet 网络号，设置网卡的 IP 地址。(如网卡 IP 地址与 subnet 网络号一致，可省略此步骤)

(5) 启动 DHCP 服务，并检测是否成功启动。

(6) 检查 Linux 防火墙与 SELinux 是否阻止了 DHCP 对外提供服务。

(7) 使用 DHCP 客户端进行测试。

4.4.2　主配置文件的作用域

dhcpd.conf 配置文件比较简单，一般由全局参数设置、全局选项设置和子网作用域设置三部分组成，具体格式如下所示：

```
全局参数设置;
全局选项设置;
子网 1 作用域定义{
     子网选项配置;
     保留主机配置{
          保留主机相关选项配置
     }
}
子网 2 作用域定义{
     子网选项配置;
     }
...
```

全局参数和全局选项对所有的子网作用域有效。当全局选项与子网选项发生冲突时，以子网选项为准。保留主机配置可以放在子网作用域里面，也可以放在子网作用域外面，根据具体操作环境而定。

4.4.3　DHCP 服务器简单配置案例

下面就配置模板中的 dhcpd.conf.sample 进行详细介绍。/etc/dhcpd.conf 文件的内容如下：

```
ddns-update-style interim;
#配置 dhcp-dns 互动更新模式
ignore client-updates;
#忽略客户端的更新
subnet 192.168.0.0 netmask 255.255.255.0 {
#定义一个 192.168.0.0/24 的子网，其中 192.168.0.0 为网络号，255.255.255.0 为子网掩码
# --- default gateway
        option routers                  192.168.0.1;
         #设置客户端网关为 192.168.0.1
        option subnet-mask               255.255.255.0;
         #设置客户端子网掩码为 255.255.255.0
        option nis-domain                "domain.org";
```

```
        #设置客户端网络信息服务器域名为 domain.org
        option domain-name              "domain.org";
        #设置客户端域名为 domain.org
        option domain-name-servers       192.168.1.1;
        #设置客户端域名服务器 IP 地址为 192.168.1.1

        option time-offset              -18000; # Eastern Standard Time
        #设置客户端时间与格林尼治时间的偏移时间为-18000 秒
#       option ntp-servers              192.168.1.1;
        #设置客户端网络时间服务器 IP 地址
#       option netbios-name-servers      192.168.1.1;
        #设置客户端 WINS 服务器 IP 地址
# --- Selects point-to-point node (default is hybrid). Don't change this unless
# -- you understand Netbios very well
#       option netbios-node-type 2;
        #设置 WINS 服务器的节点类型为 2
        range dynamic-bootp 192.168.0.128 192.168.0.254;
        #设置本子网中客户端可用地址池范围从 192.168.0.128-192.168.0.254
        default-lease-time 21600;
        #设置客户端默认租约时间为 21600 秒
        max-lease-time 43200;
        #设置客户端最大租约时间为 43200 秒
        # we want the nameserver to appear at a fixed address
        host ns {
        #设置保留主机的主机名
               next-server marvin.redhat.com;
            #设置无盘启动服务器的域名，如果不配置网络安装服务器或无盘工作站可删除此行
               hardware ethernet 12:34:56:78:AB:CD;
            #设置保留主机的网卡 MAC 地址
               fixed-address 207.175.42.254;
            #设置保留主机的固定 IP 地址
        }
        #保留主机配置符结束
}
#子网配置结束
```

4.4.4 DHCP 服务器的运行步骤

主配置文件核对无误后，在运行 DHCP 服务前还需要进行以下操作：

(1) 修改网卡启动文件/etc/sysconfig/dhcpd，设置 DHCPDARGS=eth0。

(2) 修改本机 IP 地址为 192.168.0.1/255.255.255.0。

(3) 启动 DHCP 服务：

```
# service dhcpd start
```

4.5 DHCP 客户端配置

4.5.1 在 Linux 下通过命令行配置 DHCP 客户端

在 Linux 下配置 DHCP 客户端应首先确认操作系统能够正确识别网卡。通常网卡被系

统正确识别后，会自动在/etc/sysconfig/network-scripts 目录中为网卡创建 “ifcfg-eth？” 的网卡配置文件。eth？是网络设备的名称，如 eth0、eth1 分别代表了系统中的第一块和第二块网卡。

如果要求该网卡在主机启动时能够使用 DHCP 客户端获得网络配置参数，网卡配置文件中应至少包含以下配置内容：

```
#more /etc/sysconfig/network*s/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

其中 DEVICE=eth0 指定了网卡的名称；BOOTPROTO=dhcp 表示网卡启动时利用 DHCP 自动获得地址；ONBOOT=yes 表示开机自动启动网卡。

在 CentOS 5 中，DHCP 客户端是通过 dhclient-3.0.5-29.el5_7.1 软件包实现的，通过配置 dhclient.conf 可以实现动态 DNS、别名等 DHCP 客户端功能。具体配置可使用 man dhclient.conf 命令查看手册。同时在客户端，我们可以在终端提示符下使用 dhclient 命令获取 IP 地址，如下所示：

```
# dhclient -r          //释放客户端已经获取的 IP 地址
Internet Systems Consortium DHCP Client V3.0.5-RedHat
Copyright 2004-2006 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/eth0/00:0c:29:5f:2f:58
Sending on   LPF/eth0/00:0c:29:5f:2f:58
Sending on   Socket/fallback
DHCPRELEASE on eth0 to 192.168.0.1 port 67 (xid=0x1ca98ff5)

# dhclient                //通过 dhclient 获取 DHCP 服务 IP 地址
Internet Systems Consortium DHCP Client V3.0.5-Red Hat
Copyright 2004-2006 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/eth0/00:0c:29:5f:2f:58
Sending on   LPF/eth0/00:0c:29:5f:2f:58
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3
(xid=0x542d8ece)
DHCPOFFER from 192.168.0.1
DHCPREQUEST on eth0 to 255.255.255.255 port 67 (xid=0x542d8ece)
DHCPACK from 192.168.0.1 (xid=0x542d8ece)
bound to 192.168.0.252 -- renewal in 9038 seconds.
```

在上面的命令行中，我们首先使用 dhclient -r 命令释放了 DHCP 客户端已经获得的 IP 地址，然后通过 dhclient 命令重新从 DHCP 服务器获得了 IP 地址等网络配置信息。

注意： 通过 dhclient 命令我们可以详细的看到 DHCP 客户端与服务器通信的过程，此过程与我们在 4.1.3 中所讲到的 DHCP 工作流程的 4 个过程阶段是完全一致的。

4.5.2 DHCP 客户端图形界面配置

在 CentOS 5 图形界面下，可以通过以下步骤设置 DHCP 客户端。

(1) 在 GNOME 图形界面下选择“系统”→“管理”→“网络”命令，打开如图 4-4 所示的“网络配置”对话框。可以看到列出了设备名为 eth0 的以太网卡设备。

(2) 双击 eth0 以太网卡设备，或者选中 eth0 网卡设备单击“编辑”按钮，出现图 4-5 所示的“以太网设备”配置对话框。

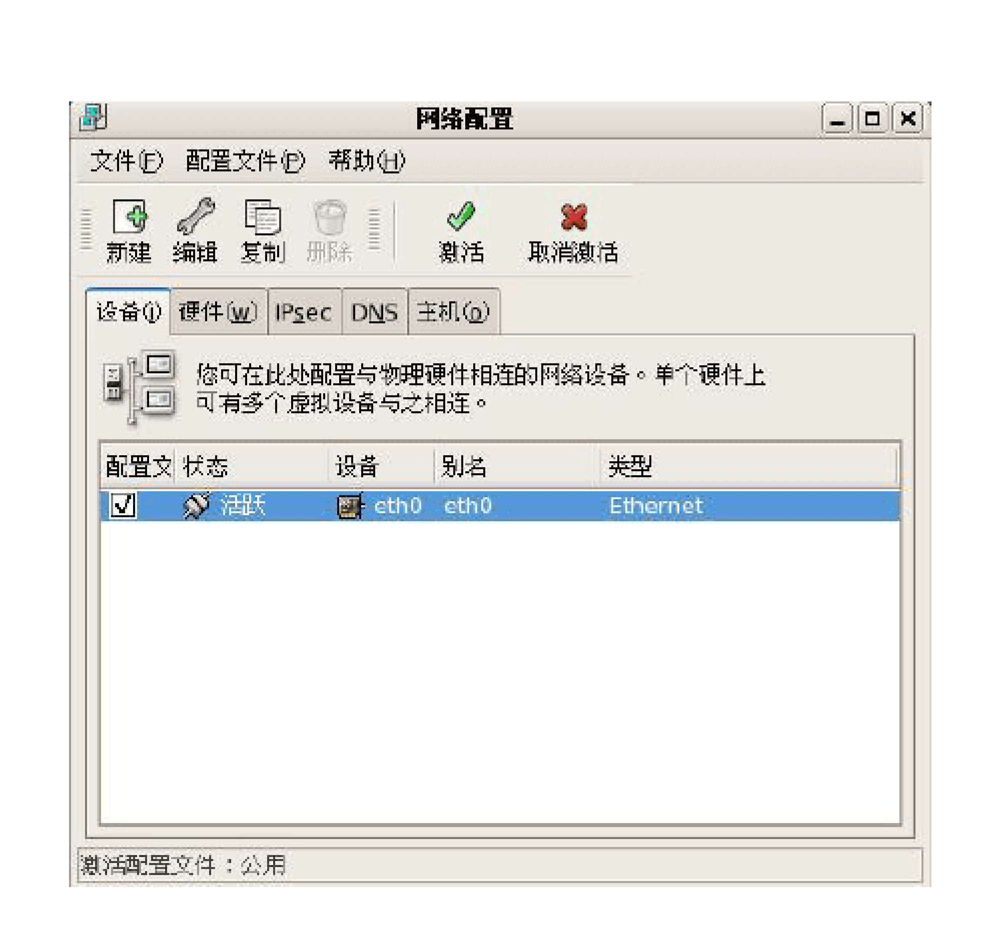


图 4-4 CentOS 5 下的“网络配置”对话框

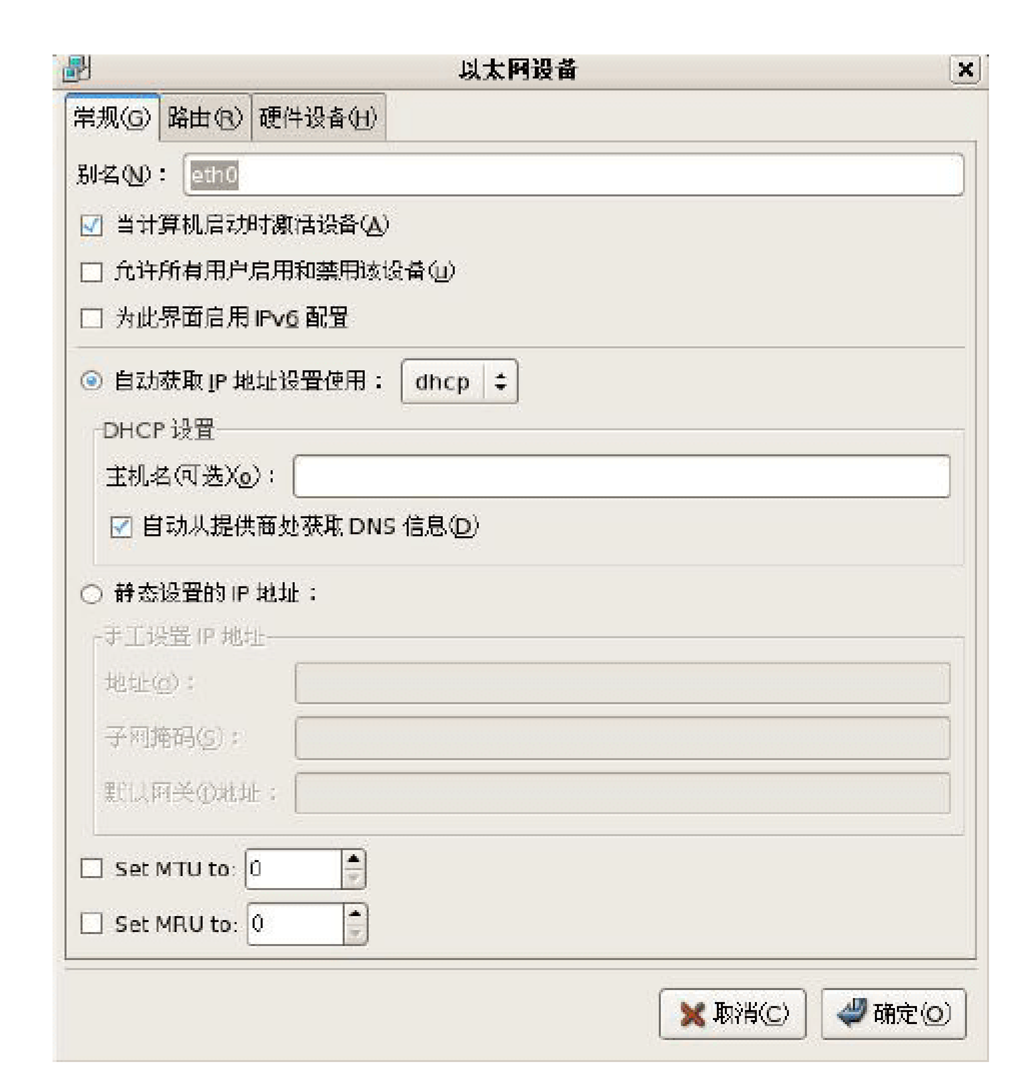


图 4-5 “以太网设备”配置对话框

(3) 选中“当计算机启动时激活设备”复选框和“自动获取 IP 地址设置使用 dhcp”单选按钮。

(4) 单击“确定”按钮，并在随后的对话框中确认保存并重新启动网络。

注意： 通过上述图形界面的操作实际上也是改变了/etc/sysconfig/network-scripts/ifcfg-eth0 的网卡配置文件。

4.5.3 Windows 下设置 DHCP 客户端

在 Windows 系统下，DHCP 的客户端配置是比较简单的。下面以 Windows 7 系统为例进行介绍。

(1) 选择“开始”→“控制面板”→“网络和共享中心”，打开图 4-6 所示的“网络和共享中心”窗口。

(2) 单击需要配置 DHCP 的网卡所对应的本地连接，打开如图 4-7 所示的“本地连接属性”对话框。

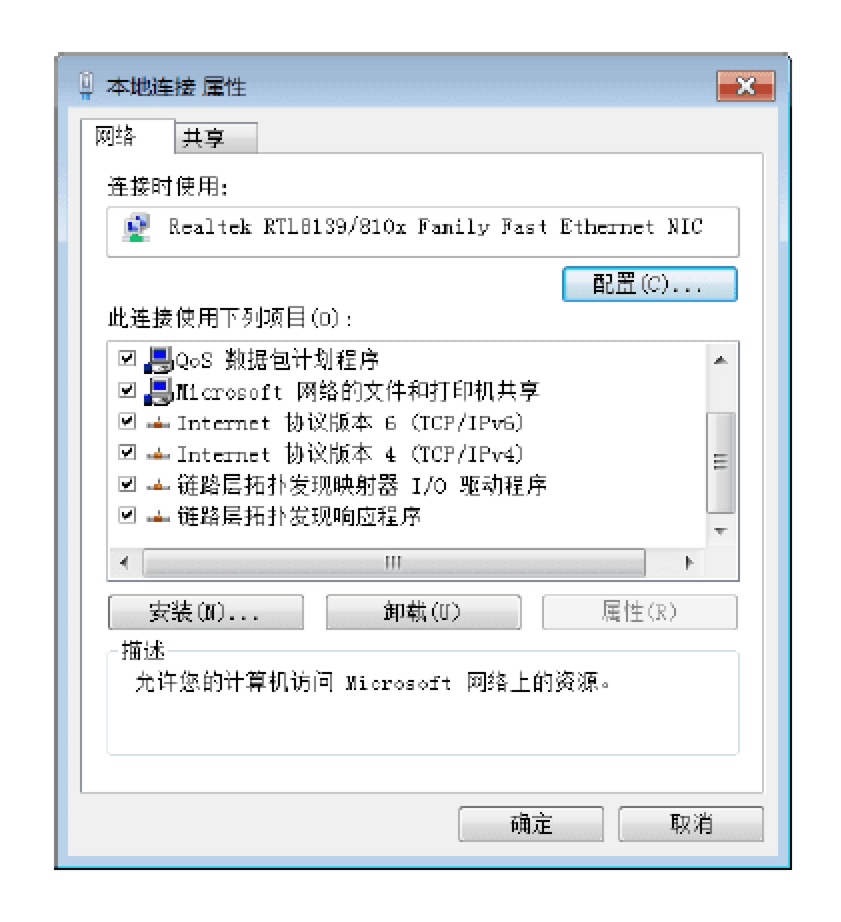
(3) 双击“Internet 协议版本 4(TCP/IPv4)”，打开 IP 地址配置对话框，如图 4-8 所示。

(4) 在如图 4-8 所示的“常规”选项卡中选中“自动获得 IP 地址”和“自动获得 DNS 服务器地址”单选按钮。

(5) 单击“确定”按钮，即可开启 Windows 的 DHCP 客户端功能。



图 4-6　Windows 7 的“网络和共享中心”窗口



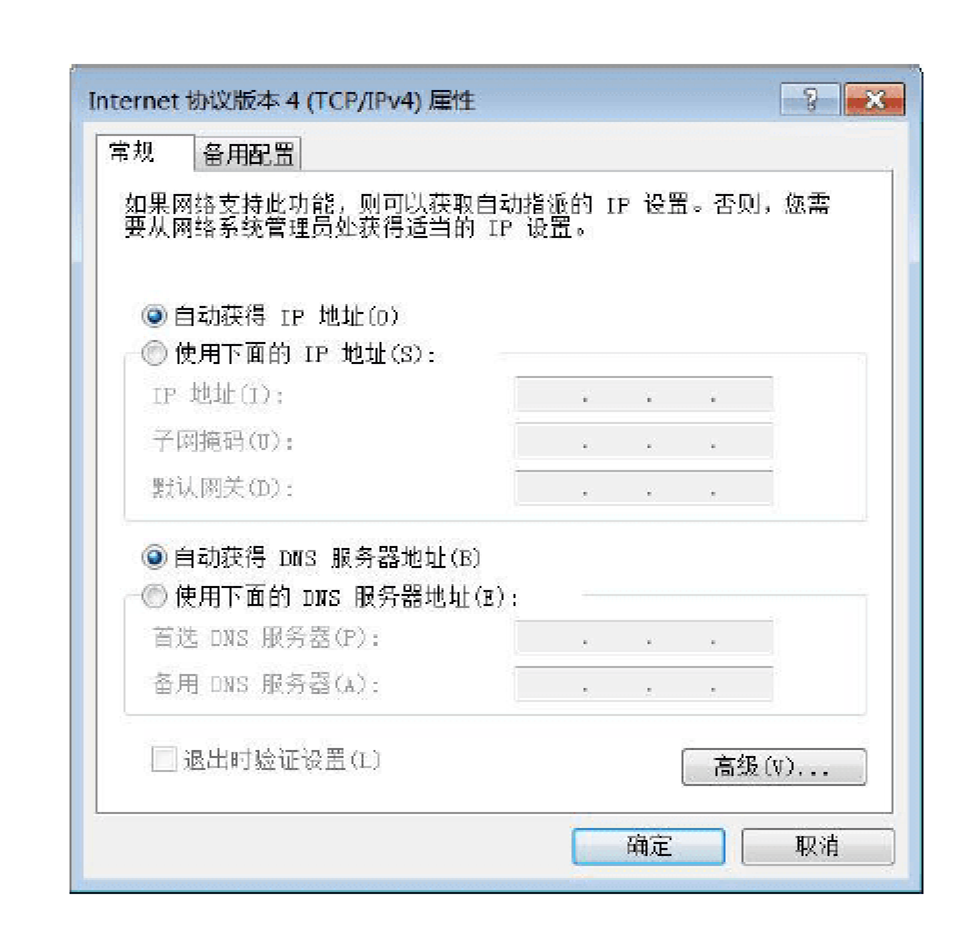


图 4-7　Windows 7 的“本地连接属性”对话框　图 4-8　“Internet 协议版本 4(TCP/IPv4)属性”对话框

4.5.4　Windows 下 DHCP 客户端命令

在 Windows 下客户端可使用 ipconfig 命令对 DHCP 情况进行查看。在 Windows 下的 DOS 提示符下可以执行以下操作。

1. 查看已经分配到的 IP 地址信息

```
ipconfig /all
```

部分运行结果如下所示：

```
以太网适配器 本地连接:

   连接特定的 DNS 后缀 . . . . . . . : domain.org
   描述. . . . . . . . . . . . . . . : Realtek RTL8139/810x Family Fast Ethernet
```

```
 NIC
   物理地址. . . . . . . . . . . . . : 00-01-03-38-16-88
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . : fe80::9444:d929:4b9b:f659%12(首选)
   IPv4 地址 . . . . . . . . . . . . : 192.168.0.251(首选)
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   获得租约的时间  . . . . . . . . . : 2012 年 3 月 19 日 21:09:26
   租约过期的时间  . . . . . . . . . : 2012 年 3 月 20 日 3:09:25
   默认网关. . . . . . . . . . . . . : 192.168.0.1
   DHCP 服务器 . . . . . . . . . . . : 192.168.0.1
   DHCPv6 IAID . . . . . . . . . . . : 251658499
   DHCPv6 客户端 DUID  . . . . . . . : 00-01-00-01-16-28-5C-B9-00-01-03-
38-16-88
```

以上结果说明该 Windows 操作系统于 2012 年 3 月 19 日 21:09:26 自 IP 地址为 192.168.0.1 的 DHCP 服务器获得 IP 地址 192.168.0.251。租约过期时间为 2012 年 3 月 20 日 3:09:25。

2. 更新当前 IP 地址信息

```
ipconfig /renew
ipconfig /all
```

部分运行结果如下所示：

```
> ipconfig /renew

Windows IP 配置
以太网适配器 本地连接:

   连接特定的 DNS 后缀 . . . . . . . : domain.org
   本地链接 IPv6 地址. . . . . . . . : fe80::9444:d929:4b9b:f659%12
   IPv4 地址 . . . . . . . . . . . . : 192.168.0.251
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   默认网关. . . . . . . . . . . . . : 192.168.0.1

> ipconfig /all

以太网适配器 本地连接:

   连接特定的 DNS 后缀 . . . . . . . : domain.org
   描述. . . . . . . . . . . . . . . : Realtek RTL8139/810x Family Fast Ethernet
 NIC
   物理地址. . . . . . . . . . . . . : 00-01-03-38-16-88
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . : fe80::9444:d929:4b9b:f659%12(首选)
   IPv4 地址 . . . . . . . . . . . . : 192.168.0.251(首选)
   子网掩码  . . . . . . . . . . . . : 255.255.255.0
   获得租约的时间  . . . . . . . . . : 2012 年 3 月 19 日 21:09:26
   租约过期的时间  . . . . . . . . . : 2012 年 3 月 20 日 3:16:45
   默认网关. . . . . . . . . . . . . : 192.168.0.1
   DHCP 服务器 . . . . . . . . . . . : 192.168.0.1
```

```
   DHCPv6 IAID . . . . . . . . . . . . . . : 251658499
   DHCPv6 客户端 DUID  . . . . . . .: 00-01-00-01-16-28-5C-B9-00-01-03-38-
16-88
```

由以上结果可知，运行过 ipconfig /renew 命令后，通过 DHCP 获得的 IP 地址和获得租约的时间都没有变化，而租约过期的时间变为 2012 年 3 月 20 日 3:16:45，即通过 ipconfig/renew 命令成功的延长了 DHCP 客户端的租约。

3. 释放已经分配到的 IP 地址信息

```
ipconfig /release
ipconfig /all
```

部分运行结果如下所示：

```
>ipconfig /release

以太网适配器 本地连接:

   连接特定的 DNS 后缀 . . . . . . . . :
   本地链接 IPv6 地址. . . . . . . . . : fe80::9444:d929:4b9b:f659%12
   默认网关. . . . . . . . . . . . . . :

>ipconfig /all

以太网适配器 本地连接:
   连接特定的 DNS 后缀 . . . . . . . . :
   描述. . . . . . . . . . . . . . . . : Realtek RTL8139/810x Family Fast Etherne
 NIC
   物理地址. . . . . . . . . . . . . . : 00-01-03-38-16-88
   DHCP 已启用 . . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . . : fe80::9444:d929:4b9b:f659%12(首选)
   自动配置 IPv4 地址  . . . . . . . . : 169.254.246.89(首选)
   子网掩码  . . . . . . . . . . . . . : 255.255.0.0
   默认网关. . . . . . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . . : 251658499
   DHCPv6 客户端 DUID  . . . . . . . . : 00-01-00-01-16-28-5C-B9-00-01-03-38-16-8

   DNS 服务器  . . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                      fec0:0:0:ffff::2%1
                                      fec0:0:0:ffff::3%1
```

以上运行结果说明本地连接的网络适配器已经释放了刚刚通过 DHCP 获得的 192.168.0.251 的 IP 地址配置信息。

4.6　DHCP 服务器配置案例

4.6.1　配置作用域案例

【例 4-1】某学校机房需要配置一台 DHCP 服务器，以满足机房内机器上网需求。所有机器全部采用 DHCP 动态 IP 地址实现。其中网关为 10.16.1.254，DNS 服务器域名为

ns.edu.cn，DNS 服务器 IP 地址为 10.16.1.1，分配的地址池为 10.16.1.100-10.16.1.253。具体实施步骤如下。

(1) 创建 DHCP 主配置文件，直接从模板中复制：

```
#copy /usr/share/doc/dhcp*/dhcpd.conf.sample  /etc/dhcpd.conf
```

(2) 根据上述要求，修改/etc/dhcpd.conf 主配置文件内容，修改后内容如下：

```
ddns-update-style interim;
ignore client-updates;
subnet 10.16.1.0 netmask 255.255.255.0 {
# 定义 10.16.1.0/24 子网作用域
       option routers                10.16.1.254;
        #设置网关为 10.16.1.254
       option subnet-mask             255.255.255.0;
        #设置子网掩码为 255.255.255.0
       option domain-name             "ns.edu.cn";
        #设置域名为 ns.edu.cn
       option domain-name-servers      10.16.1.1;
        #设置 DNS 服务器 IP 地址为 10.16.1.1
       option time-offset             -18000; # Eastern Standard Time
       range dynamic-bootp 10.16.1.100 10.16.1.253;
        #设置地址池为 10.16.1.100 到 10.16.1.253
       default-lease-time 21600;
        #设置默认租期 21600 秒
       max-lease-time 43200;
        #设置最大租期 43200 秒
}
```

(3) 配置 DHCP 网卡启动接口，修改/etc/sysconfig/dhcpd 配置文件，修改内容如下：

```
DHCPDARGS="eth0"
```

(4) 配置 DHCP 服务器网卡地址，修改网卡/etc/sysconfig/network-scripts/ifcfg-eth0 配置文件如下：

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=10.16.1.255
IPADDR=10.16.1.1
NETMASK=255.255.255.0
NETWORK=10.16.1.0
ONBOOT=yes
GATEWAY=10.16.1.254
TYPE=Ethernet
```

(5) 启动 DHCP 服务：

```
# service network restart       //重新配置网络
# service dhcpd start           //启动 DHCP 服务
# netstat -anup | grep dhcp     //查看 DHCP 是否启动成功，若 UDP  67 的 DHCP 端口
启动成功，说明 DHCP 启动成功。
```

(6) 利用 DHCP 客户端进行测试。在 Linux 客户端输入 dhclient 命令，检测 DHCP 客户端是否分配到正确的网络配置信息。

```
dhclient
```

注意： DHCP 服务配置比较简单，但要注意以下几点：

① 网卡 IP 地址需要在主配置文件定义的子网作用域的网段范围内。

② 如果/var/lib/dhcp/dhcpd.leases 租期文件不存在，启动 DHCP 时也会提示错误，手工建立此文件即可。

③ 主配置文件中地址池网段应与作用域网段一致，即地址池中的 IP 地址应包含在作用域网段中。

4.6.2　配置子网作用域案例

在某些学校，各部门的 IP 地址是严格分开的，各个部门使用不同网段的 IP 地址。但可能各部门人数又不是很多，此时可采用 DHCP 服务器子网作用域的方式进行配置。为了更好地利用 IP 地址，需要将原有的网段划分成多个子网。

【例 4-2】学校办公室有 20 台计算机，使用笔记本电脑接入的员工数目最多时能达到 5 人，需要配置一个 DHCP 服务器，采用 192.168.1.0 网段。网络管理员为节约 IP 地址为办公室分配了 30 个 IP 地址。请根据其实际情况为办公室架设一台 DHCP 服务器。

(1) 根据上述需求可以知道，办公室可用的 IP 地址在 30 个，由此可确定办公室可使用的 IP 为 192.168.1.1～192.168.1.30。由此可以确定，该网络的子网掩码为 255.255.255.224，网段标识为 192.168.1.0，广播地址为 192.168.1.31。

(2) 修改主配置文件，修改后内容如下：

```
ddns-update-style interim;
ignore client-updates;
subnet 192.168.1.0 netmask 255.255.255.224 {
# 定义 192.168.1.0/24 子网作用域
        option routers                  192.168.1.254;
         #设置网关为 192.168.1.254
        option subnet-mask               255.255.255.0;
         #设置子网掩码为 255.255.255.0
        option domain-name               "ns.edu.cn";
         #设置域名为 ns.edu.cn
        option domain-name-servers        10.16.1.1;
         #设置 DNS 服务器 IP 地址为 10.16.1.1
        option time-offset               -18000; # Eastern Standard Time
        range dynamic-bootp 192.168.1.2 192.168.1.30;
         #设置地址池为 192.168.1.2 到 192.168.1.30
        default-lease-time 21600;
         #设置默认租期 21600 秒
        max-lease-time 43200;
         #设置最大租期 43200 秒
```

(3) 配置 DHCP 服务器 IP 地址，修改/etc/sysconfig/network-scripts/ifcfg-eth0 配置文件如下：

```
DEVICE=eth0
BOOTPROTO=static
IPADDR=192.168.1.1
NETMASK=255.255.255.0
ONBOOT=yes
GATEWAY=192.168.1.254
TYPE=Ethernet
```

(4) 配置 DHCP 服务启动接口，修改/etc/sysconfig/dhcpd 配置文件如下：

```
DHCPDARGS="eth0"
```

(5) 启动 DHCP：

```
# service network restart       //重新配置网络
# service dhcpd start           //启动 DHCP 服务
# netstat -anup | grep dhcp     //查看 DHCP 是否启动成功，若 UDP 67 的 DHCP 端口
启动成功，说明 DHCP 启动成功。
```

(6) 利用 DHCP 客户端进行测试。在 Linux 客户端输入 dhclient 命令，检测 DHCP 客户端是否分配到正确的网络配置信息。

```
dhclient
```

4.6.3 配置多作用域网络案例

DHCP 服务器支持配置多作用域，使得 DHCP 服务器能为多个网络提供 IP 地址自动分配的服务。

【例 4-3】现学校有两个机房，机房 1 所使用地址池为 10.16.1.0/24，网关为 10.16.1.254，DNS 服务器 IP 地址为 10.16.1.1，域名为 ns1.edu.cn。机房 1 中保留主机为 WWW 服务器，IP 地址为 10.16.1.100，MAC 地址为 11-22-33-44-55-66。机房 2 所使用的地址池为 10.16.2.0/24，网关为 10.16.2.254，DNS 服务器为 10.16.2.1，域名为 ns2.edu.cn。现有一台 DHCP 服务器以太网 eth0 口与机房 1 网络相连，以太网 eth1 口与机房 2 网络相连，如图 4-9 所示。请配置这台 DHCP 服务器。

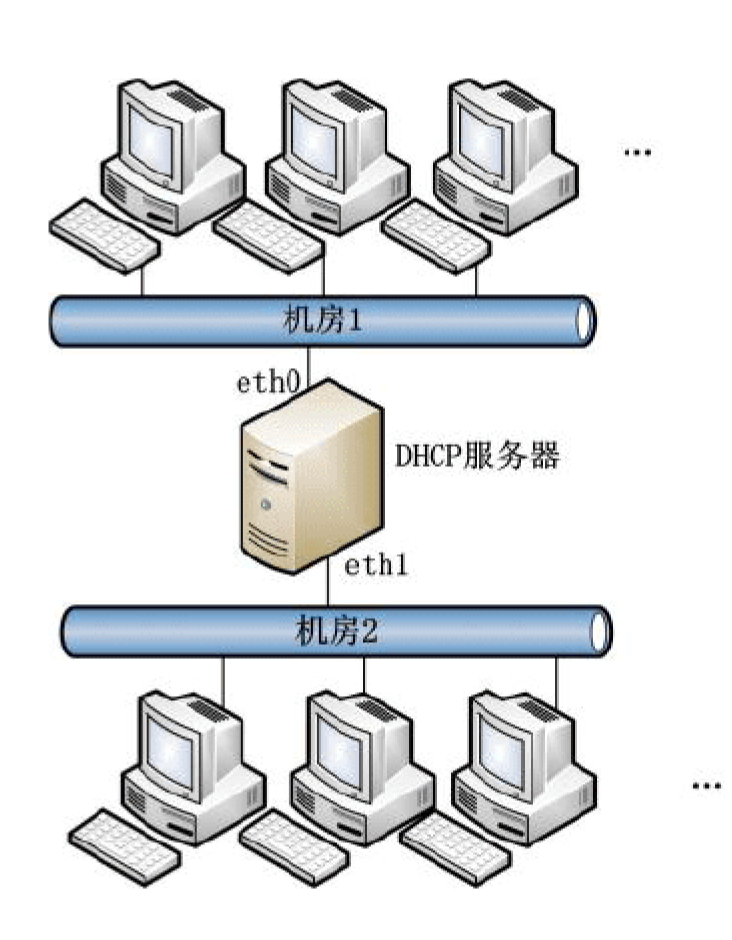


图 4-9 DHCP 多作用域拓扑结构

具体实施步骤如下。

(1) 配置 DHCP 服务器，修改主配置文件如下：

```
ddns-update-style interim;
ignore client-updates;
subnet 10.16.1.0 netmask 255.255.255.0 {
# 定义机房 1 的 10.16.1.0/24 子网作用域
        option routers                 10.16.1.254;
         #设置网关为 10.16.1.254
        option subnet-mask              255.255.255.0;
         #设置子网掩码为 255.255.255.0
        option domain-name              "ns1.edu.cn";
         #设置域名为 ns1.edu.cn
        option domain-name-servers       10.16.1.1;
         #设置 DNS 服务器 IP 地址为 10.16.1.1
        option time-offset              -18000; # Eastern Standard Time
        range dynamic-bootp 10.16.1.2 10.16.1.253;
         #设置地址池为 10.16.1.2 到 10.16.2.253
        default-lease-time 21600;
```

```
        #设置默认租期 21600 秒
       max-lease-time 43200;
        #设置最大租期 43200 秒
       host WWW{
        #设置 www 主机
        hardware ethernet 11:22:33:44:55:66;
        fixed-address 10.16.1.100;
        #将 MAC 地址为 11:22:33:44:55:66 的主机分配固定 ip：10.16.1.100
        }
}
subnet 10.16.2.0 netmask 255.255.255.0 {
# 定义机房 2 的 10.16.2.0/24 子网作用域
       option routers                10.16.2.254;
        #设置网关为 10.16.2.254
       option subnet-mask             255.255.255.0;
        #设置子网掩码为 255.255.255.0
       option domain-name             "ns2.edu.cn";
        #设置域名为 ns2.edu.cn
       option domain-name-servers      10.16.2.1;
        #设置 DNS 服务器 IP 地址为 10.16.2.1
       option time-offset             -18000; # Eastern Standard Time
       range dynamic-bootp 10.16.2.2 10.16.2.253;
        #设置地址池为 10.16.2.2 到 10.16.2.253
       default-lease-time 21600;
        #设置默认租期 21600 秒
       max-lease-time 43200;
        #设置最大租期 43200 秒
}
```

(2) 设置 eth0 网卡的 IP 地址，编辑/etc/sysconfig/network-scripts/ifcfg-eth0 修改如下：

```
DEVICE=eth0
BOOTPROTO=static
IPADDR=10.16.1.1
NETMASK=255.255.255.0
ONBOOT=yes
GATEWAY=10.16.1.254
TYPE=Ethernet
```

(3) 设置 eth1 网卡的 IP 地址，编辑/etc/sysconfig/network-scripts/ifcfg-eth1 修改如下：

```
DEVICE=eth1
BOOTPROTO=static
IPADDR=10.16.2.1
NETMASK=255.255.255.0
ONBOOT=yes
GATEWAY=10.16.2.254
TYPE=Ethernet
```

(4) 开启路由转发功能，使之立刻生效，并添加相应路由如下：

```
# echo "1" /proc/sys/net/ipv4/ip_forward
# sysctl -p
# route add -host 255.255.255.255 dev eth0
```

(5) 设置 DHCP 服务器启动接口，修改/etc/sysconfig/dhcpd 配置文件如下：

```
DHCPDARGS="eth0,eth1"
```

(6) 重启 network 服务，重启 DHCP 服务：

```
# service network restart        //重新配置网络
# service dhcpd start            //启动 DHCP 服务
# netstat -anup | grep dhcp      //查看 DHCP 是否启动成功，若 UDP 67 的 DHCP 端口
启动成功，说明 DHCP 启动成功。
```

(7) 在机房 1 中 www 主机上面测试 DHCP 服务：

```
dhclient
```

(8) 在机房 1 中其他主机上测试 DHCP 服务：

```
dhclient
```

(9) 在机房 2 中主机上面测试 DHCP 服务：

```
dhclient
```

4.6.4 配置保留主机与保留主机组案例

在构架 DHCP 服务器时，需要考虑为某些特定的服务器分配保留静态的 IP 地址，以防止服务器 IP 发生动态更新的情况发生。例如网络中的文件服务器、邮件服务器、DNS 服务器、主页服务器等，都需要为其分配一个特定的保留地址。

保留主机的基本语句如下：

```
host hostname{
#定义保留主机的主机名
        option 选项;
        #设置保留主机的客户端选项，如网关、域名、DNS 服务器 IP 等
        hardware ethernet MAC 地址;
        #设置保留主机的 MAC 地址
        fixed-address IP 地址;
        #设置保留主机的 IP 地址
        }
```

在例 4-3 中，我们定义了一台保留主机，如果网络中存在多台保留主机，可以设置多个与以上语句类似的内容。

保留主机可以设置在子网里面，成为子网的保留主机；也可以设置在子网外面，成为一个独立的保留主机组。将同一类的保留主机定义在一个组中，即方便管理员管理，又方便管理员查阅。

保留主机组的基本语句如下：

```
group{
    host hostname1{
    #定义组中保留主机的主机名
    option 选项;
    #定义保留主机的客户端选项如网关、DNS 等
    hardware ethernet MAC 地址;
    #设置保留主机的 MAC 地址
    fixed-address IP 地址;
    #设置保留主机的 IP 地址
```

```
        }
        host hostname2{
        option 选项;
        hardware ethernet MAC 地址;
        fixed-address IP 地址;
        }
    ...

    }
```

【例 4-4】在机房 1 中安装配置了多台服务器：FTP 服务器 MAC 地址为“00:11:22:33:44:55”，主机名为 ftp，IP 地址为 10.16.1.2；WWW 服务器 MAC 地址为“11:22:33:44:55:66”，主机名为 www，IP 地址为 10.16.1.1；MAIL 服务器 MAC 地址为“22:33:44:55:66:77”，主机名为 mail，IP 地址为 10.16.1.3。分别设置保留主机与保留主机组。

(1) 若设置保留主机，则/etc/dhcpd.conf 主配置文件中，应设置内容如下：

```
host ftp{
 hardware ethernet 00:11:22:33:44:55;
 fixed-address 10.16.1.2;
 }
host www{
 hardware ethernet 11:22:33:44:55:66;
 fixed-address 10.16.1.1;
 }
host mail{
 hardware ethernet 22:33:44:55:66:77;
 fixed-address 10.16.1.3;
 }
```

(2) 若设置保留主机组，则/etc/dhcpd.conf 主配置文件中，应设置内容如下：

```
group{
        host ftp{
         hardware ethernet 00:11:22:33:44:55;
         fixed-address 10.16.1.2;
         }
        host www{
         hardware ethernet 11:22:33:44:55:66;
         fixed-address 10.16.1.1;
         }
        host mail{
         hardware ethernet 22:33:44:55:66:77;
         fixed-address 10.16.1.3;
         }
}
```

4.6.5　配置 DHCP 中继代理服务器

一般情况下，DHCP 请求的广播包是不能通过路由器的，因为路由器具有隔离广播的功能。如果在两个子网中只构架一个 DHCP 服务器，DHCP 服务器处于局域网 1，局域网 2 的客户将无法正常从 DHCP 服务器获取 IP 地址。为了更好地解决这个问题，需要在局域网 2 中架设一台 DHCP 中继代理服务器。

【例 4-5】机房内有两个网段，局域网 1 的 IP 地址范围是 192.168.1.0/24，局域网 2 的地址范围是 192.168.2.0/24。现局域网 1 内有一台 DHCP 服务器，IP 地址是 192.168.1.1，要求在局域网 2 中假设一台 DHCP 中继代理服务器。当局域网 2 中的 DHCP 客户端需要发送 DHCP 请求时，DHCP 中继代理在接收到请求包后以客户端的身份向局域网 1 的 DHCP 服务器发起请求，并将请求的结果返回给 DHCP 客户端，以协助完成 DHCP 客户端 IP 地址的请求，如图 4-10 所示。

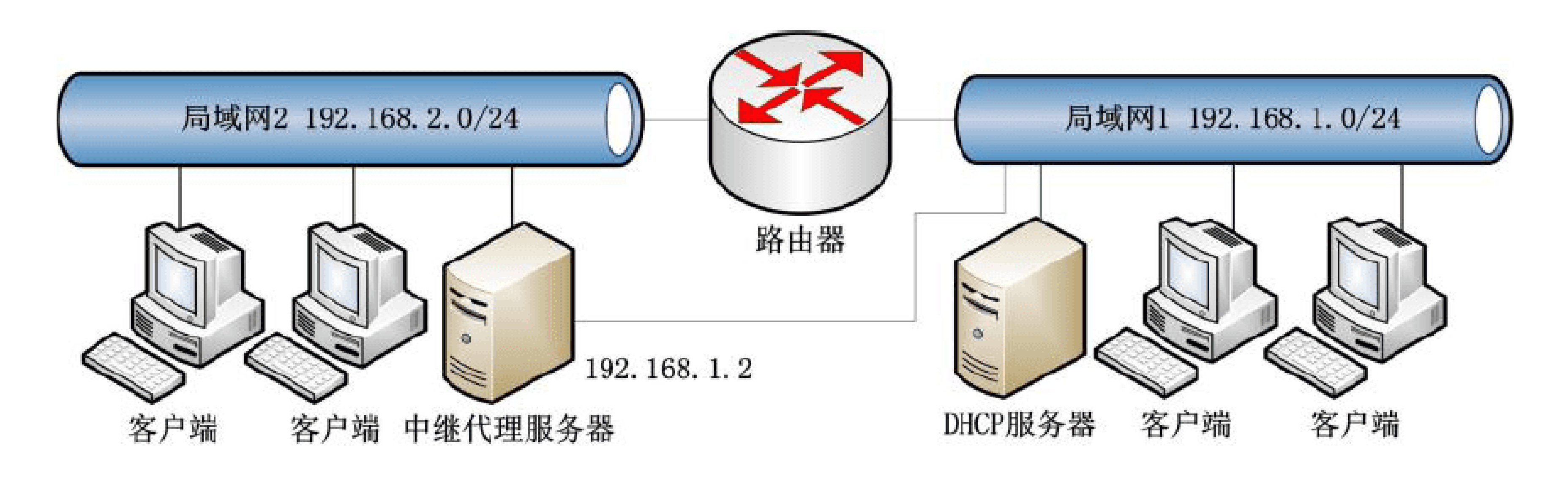


图 4-10 中继代理服务器拓扑

DHCP 中继代理服务器既然起到中继代理的功能，因而就需要一个静态的 IP 地址，同时还需要知道 DHCP 服务器的 IP 地址，以及通过哪个接口向 DHCP 服务器发送请求。假如局域网 2 的 DHCP 中继代理服务器的 IP 地址是 192.168.2.253，局域网 1 的 DHCP 服务器 IP 地址为 192.168.1.1，现需要配置中继代理服务器，具体配置如下：

(1) 修改 DHCP 中继代理服务器，设置 DHCP 服务器 IP 地址及发送请求的接口。修改 DHCP 中继代理配置文件/etc/sysconfig/dhcrelay 内容如下：

```
INTERFACE="eth0"
DHCPSERVERS="192.168.1.1"
```

(2) 开启路由功能，启动 DHCP 中继代理服务：

```
# echo "1" /proc/sys/net/ipv4/ip_forward
# sysctl -p
# route add -host 255.255.255.255 dev eth0

service dhcrelay start
```

(3) 使用 netstat 检测端口是否已经打开

```
netstat -anup | grep dhcrelay
```

显示结果如下：

```
udp      0       0    0.0.0.0:67       0.0.0.0:*       3519/dhcrelay
```

在对 DHCP 中继代理服务器进行配置时，需要知道 DHCP 服务器的 IP 地址。DHCP 中继代理服务器之所以开放 67 端口，是因为 DHCP 客户端在发送请求时，其目标端口为 67 号端口。而 DHCP 客户端无法判断接收这个请求广播包的是 DHCP 服务器还是 DHCP 中继代理，所以 DHCP 服务器的端口号与中继代理服务器的端口号是一致的。

注意： 除非使用 INTERFACES 指令在/etc/sysconfig/dhcprelay 文件中制定接口，DHCP 中继代理默认监听所有接口上的 DHCP 请求。

4.7 本 章 小 结

本章介绍了在 Linux 中架设 DHCP 服务器的具体方法。首先介绍了 DHCP 的原理、优点及相关术语；之后对 DHCP 服务器的安装、运行方法及常用的 DHCP 配置文件及选项进行了详细介绍；接着介绍了 DHCP 服务器与 DHCP 客户端的具体配置步骤和方法；最后用 5 个案例具体详尽地说明了 5 种不同类型 DHCP 服务器的配置方法。

DHCP 是网络中最常见、最基本的应用，通过本章介绍的操作，读者可自行在 Linux 系统中架设 DHCP 服务器，为局域网中的主机提供动态 IP 地址服务。

4.8 课 后 练 习

1. 填空题

(1) DHCP 的全称是__________。

(2) DHCP 服务器的主要功能是动态分配__________。

(3) DHCP 服务器安装好后并不是立即就可以给 DHCP 客户端提供服务，它必须经过一个__________步骤。未经此步骤的 DHCP 服务器在接收到 DHCP 客户端索取 IP 地址的要求时，并不会给 DHCP 客户端分派 IP 地址。

(4) 如果要设置保留 IP 地址，则必须把 IP 地址和客户端的__________进行绑定。

2. 选择题

(1) DHCP 服务器能提供给客户机(　　)配置。

A. IP 地址　　B. 子网掩码　　C. 默认网关　　D. DNS 服务器

(2) DHCP 客户端的租约文件默认保存在(　　)目录下。

A. /etc/dhcpd　　B. /var/log/dhcpd　C. /var/lib/dhcp/　D. /var/lib/dhcpd/

(3) DHCP 是动态主机配置协议的简称，其作用是可以使网络管理员通过一台服务器来管理一个网络系统，自动地为一个网络中的主机分配(　　)地址。

A. 网络　　B. MAC　　C. TCP　　D. IP

(4) 为保证在启动服务器时自动启动 DHCP 进程，应对(　　)文件进行编辑。

A. /etc/rc.d/rc.inet2　　B. /etc/rc.d/rc.inet1

C. /etc/dhcpd.conf　　D. /etc/rc.d/rc.S

(5) 下列(　　)参数用于定义 DHCP 服务地址池。

A. host　　B. range　　C. ignore　　D. subnet

(6) DHCP 客户端在广播 IP 租约请求时使用的端口(　　)。

A. TCP 67　　B. TCP 68　　C. UDP 67　　D. UDP 68

(7) 以下属于 DHCP 租约文件的是(　　)。

A. /var/lib/dhcpd/ dhcpd.leases　　B. /var/lib/dhcp/ dhcpd.leases

C. / usr/lib/dhcpd/ dhcpd.leases　　D. /etc/lib/dhcp/ dhcpd.leases

(8) DHCP 服务器默认启动脚本(　　)。

A. dhcpd B. dhcp C. dhclient D. network

(9) 以下属于广播消息的有( )。

A. DHCPDISCOVER B. DHCPOFFER

C. DHCPREQUEST D. DHCPACK

3. 简答题

(1) 简述 DHCP 的工作过程。

(2) 简述 DHCP 的优缺点。

(3) 认真阅读以下说明信息，回答问题。

【说明】

在一个基于 TCP/IP 协议的网络中，每台主机都有一个 IP 地址，根据获得 IP 地址方式的不同，可以分为静态 IP 和动态 IP。例如：用宽带入网，会有一个固定的 IP 地址，每次连入 Internet，你的 IP 都一样；而用拨号上网，每次连入 Internet 时都从 ISP 那里获得一个 IP 地址且每次获得的可能都不同，这是因为 DHCP 服务器的存在。在 Linux 中建立 DHCP 服务器的配置文件是“dhcpd.conf ”,每次启动 DHCP 服务器都要读取该文件。下面是一个 dhcp.conf 文件的实例：

```
1 default-lease-time 1200;
2 max-lease-time 9200;
3 option subnet-mask 255.255.255.0;
4 option broadcast-address 192.168.1.255;
5 option router 192.168.1.254;
6 option domain-name-serves 192.168.1.1, 192.168.1.2
7 option domain-name “abc.com”
8 subnet 192.168.1.0 netmask 255.255.255.0
9 {
10 range 192.168.1.20 192 .168.1.200;
11 }
12 host fixed{
13 option host-name “ fixed.abc.com”;
14 hardware Ethernet 00:A0:78:8E:9E:AA;
15 fixed-address 192.168.1.22;
16 }
```

① 该 DHCP 服务器可分配的 IP 地址有多少个？

② 该 DHCP 服务器指定的默认网关、域名以及指定的 DNS 服务器分别是什么？

③ 该配置文件的 12～15 行实现什么配置功能？

④ Windows 操作系统下通过什么命令可以知道本地主机当前获得的 IP 地址？

第 5 章

NFS 服务的配置及应用

1984 年 Sun(Sun Microsystems，现已被 Oracle 收购)公司为了让不同计算机、不同操作系统之间可以彼此共享文件而开发出 NFS。由于 NFS 使用起来非常方便，因此很快得到了大多数的 UNIX/Linux 系统的广泛支持，并逐渐普及起来。目前已经成为 IETE(国际互联网工程组)制定的 RFC1904、RFC1813 和 RFC3010 文件服务标准。本章将全面地介绍 NFS 服务的安装、配置、管理和使用。

5.1 NFS 服务简介

5.1.1 NFS 概述

NFS 即网络文件系统(Network File System)，是使不同的计算机之间能通过网络进行文件共享的一种网络协议，主要用于 UNIX/Linux 操作系统中。NFS 可以与很多服务结合起来一起使用，可以提高服务的性能、安全性及数据传输速率。NFS 服务还可以满足多人同时对共享目录进行操作的需求，能够自动保持数据的一致性。

NFS 网络文件系统支持应用程序在客户端通过网络存取位于服务器磁盘中的数据。NFS 的基本原则是让不同的客户端及服务器通过一组 RPCs 共享相同的文件系统，它独立于操作系统，允许不同硬件及操作系统共同进行文件的共享。这样的好处就是除了提升资源的使用效率，还可以大大节省硬盘的空间，因为每台主机不需要将所有的文件都存储到本地硬盘上，同时也可以做到资源集中管理。

一台 NFS 服务器就如同一台文件服务器，只要将其文件系统共享出来，NFS 客户端就可以将它挂载到本地系统中，从而可以像使用本地文件系统中的文件一样使用那些远程文件系统中的文件。

NFS 采用客户/服务器工作模式。如图 5-1 所示，在 NFS 服务器上将/nfs/public 目录设置为输出目录(即共享目录)后，其他客户端就可以将这个目录挂载到自己系统中的某个目录下，这个目录可以与服务器上的输出目录和其他客户机中的目录不相同，如图中的客户机 PC1 与 PC2 的挂载目录就不相同。如果某用户登录到客户机 PC1 并进入/mnt/share 目录，那么就可以看到 NFS 服务器内/nfs/public 目录下的所有子目录及文件，只要具有相应的权限，就可以对磁盘或文件进行相应的操作。

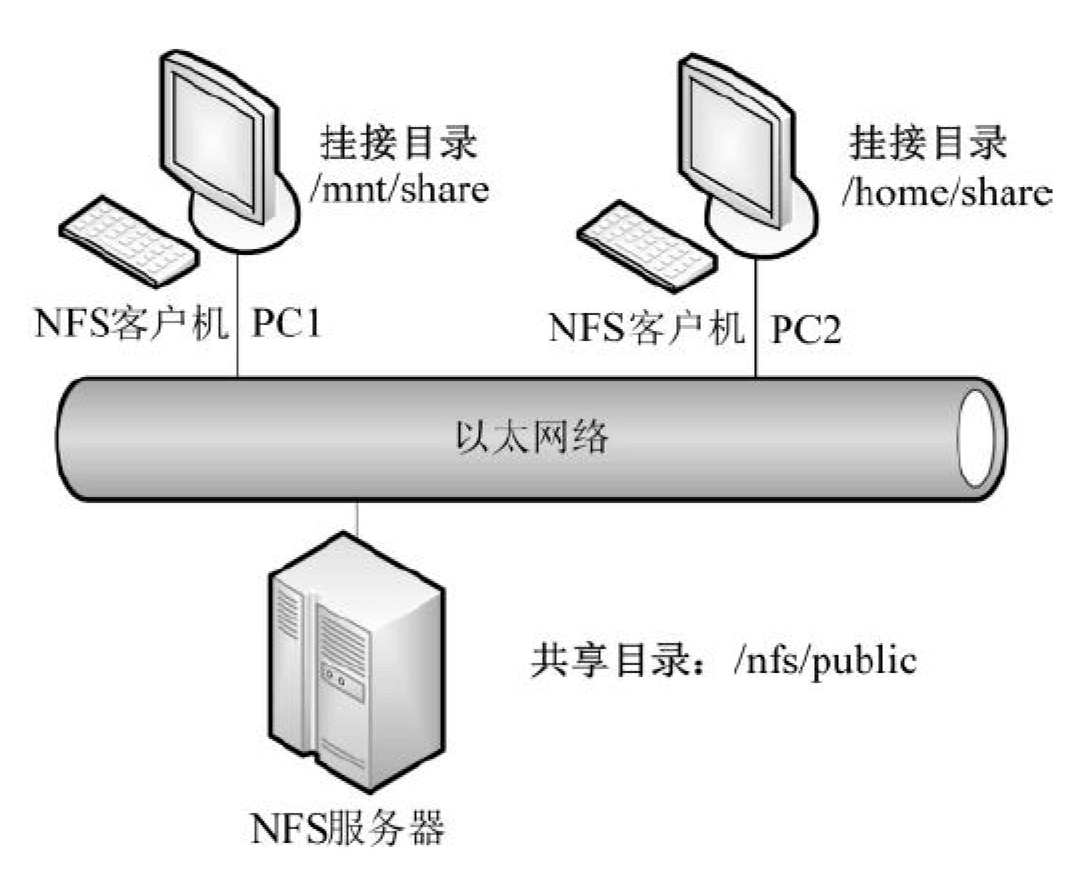


图 5-1 NFS 服务器输出目录与客户机挂载示意图

5.1.2 NFS 的优势

NFS 对于同一网络中的多个 Linux 用户间共享目录很有优势。例如，从事同一项目的工程师可以将工作目录通过 NFS 服务器进行共享，客户端可将服务器上的项目工作目录

挂载到本机的某个挂接点上。客户端挂载成功后，工程师可以像使用本地硬盘目录一样使用工作目录，不需要输入口令，也不需要使用特殊命令，存取共享文件十分方便。

NFS 客户端对于共享目录中文件或内容的修改将直接改变 NFS 服务器的共享目录，即 NFS 客户端与服务器对文件的共享是实时同步的，解决了使用 FTP 服务器文件同步困难的问题。

通过将常用数据存储在 NFS 服务器上，NFS 可以让网络中服务器的磁盘空间得到较高的利用率，让本地工作站使用更少的空间。另外，使用 NFS 还可以实现共享移动介质设备，如 DVD-ROM 等，节省硬件投资。

5.1.3　NFS 工作流程

虽然 NFS 可以在网络中进行文件共享，但 NFS 在设计时并没有提供数据传输的功能，因此，它需借助 RPC(Remote Procedure Calls，远程过程调用)。RPC 定义了一种进程间通过网络进行交互通信的机制，它允许客户端进程通过网络向远程服务进程请求服务，而不需要了解服务器底层通信协议的详细信息。

在一个 RPC 连接建立开始阶段，客户端建立过程调用(Procedure Call)，将调用参数发送到远程服务器进程，并且等待响应。当请求到达时，服务器通过客户端请求的服务调用指定的程序，并将结果返回客户端。当 RPC 调用结束，客户端程序将继续进行余下的通信操作。

NFS(v2、v3)依赖 RPC 与外部通信，为了保证 NFS 服务正常工作，其需要在 RPC 注册相应的服务端口信息，这样客户端向服务器的 RPC 提交访问某个服务的请求时，服务器才能够正确作出响应。

注册 NFS 服务时，需要先开启 RPC，才能保证 NFS 注册成功。并且如果 RPC 服务重新启动，其保存的信息会丢失，需要重新启动 NFS 服务进程，以注册端口信息，否则客户端将无法访问 NFS 服务器。

使用 NFS 服务，至少需要启动以下 3 个系统守护进程。

1) rpc.nfsd

它是基本的 NFS 守护进程，主要功能是管理客户端是否能够登入服务器。

2) rpc.mountd

它是 RPC 安装守护进程，主要功能是管理 NFS 的文件系统。当客户端顺利地通过 rpc.nfsd 登录 NFS 服务器后，在使用 NFS 服务器所提供的文件前，还必须通过文件使用权限的验证，rpc.mountd 会读取 NFS 的配置文件/etc/exports 来对比客户端的权限。

3) portmap

portmap 的主要功能是进行端口映射工作。当客户端尝试连接并使用 RPC 服务器提供的服务(如 NFS 服务)时，portmap 会将所管理的与服务对应的端口号提供给客户端，从而使客户端可以通过该端口向服务器请求服务。

值得注意的是，虽然 portmap 只用于 RPC，但它对 NFS 服务来说是必不可少的。portmap 没有运行，NFS 客户端就无法查找从 NFS 服务器中共享的目录。

5.2 NFS 服务的安装与运行

5.2.1 安装 NFS 服务

目前几乎所有的 Linux 发行版都默认安装了 NFS 服务。CentOS 5 中只要按照默认配置安装了系统，NFS 服务就已经安装在系统中。严格意义上来说，NFS 需要 5 组 RPM 包，它们分别是：

- setup-*：共享 NFS 目录在/etc/exports 中定义。
- initscripts-*：包括引导过程中装载网络目录的基本脚本。
- nfs-utils-*：包括基本的 NFS 命令与监控程序。
- portmap-*：支持安全 NFS、RPC 服务的连接。
- quota-*：网络上共享的目录配额，包括 rpc.rquotad (这个包不是必须的)。

启动 NFS 服务时需要 nfs-utils 和 portmap 这两个软件包，因此在配置使用 NFS 之前，可使用下面的命令来检查系统中是否已经安装了这两个包。

```
rpm -qa | grep nfs
rpm -qa | grep portmap
```

命令执行结果如下所示：

```
 [root@ha01 /]# rpm -qa | grep nfs
nfs-utils-1.0.9-44.el5
nfs-utils-lib-1.0.8-7.6.el5

[root@ha01 /]# rpm -qa | grep portmap
portmap-4.0-65.2.2.1
```

由上可见，系统当前已经安装了 NFS 服务和 portmap 服务。如果系统尚未安装这两项服务，则可将 CentOS 的安装盘放入光驱。加载光驱后，在光盘的 CentOS 目录下运行如下命令进行安装：

```
rpm -ivh setup-* initscripts-* nfs-utils-* portmap-* quota-* --force --
noscripts
```

安装结果显示如下：

```
[root@localhost CentOS]# rpm -ivh setup-* initscripts-* nfs-utils-*
portmap-* quota-* --force --noscripts
Preparing...                ####################################### [100%]
   1:setup                 ######################################## [ 14%]
   2:nfs-utils-lib          ####################################### [ 29%]
   3:initscripts           ######################################## [ 43%]
   4:nfs-utils-lib-devel    ####################################### [ 57%]
   5:portmap               ######################################## [ 71%]
   6:nfs-utils             ######################################## [ 86%]
   7:quota                 ######################################## [100%]
```

可见 nfs 服务已经成功安装。在某些 CentOS 5 版本中，用户手工安装 nfs 服务可能会出现脚本执行错误的情况，因此在使用 rpm 安装时可加入--noscripts 参数表示不执行脚本

内容。如已经安装了 nfs 需要强制安装，可加入--forece 参数。

5.2.2　启动 NFS 服务

在 5.1.3 节中讲到，NFS 启动需要启动三个守护进程，分别是 portmap、rpc.nfsd、rpc.mountd。下面介绍启动 NFS 服务方法。

1) 启动 portmap 守护进程

```
service portmap start
```

显示结果如下：

```
[root@localhost CentOS]# service portmap start
启动 portmap:                                             [确定]
```

2) 启动 NFS 服务

```
service nfs start
```

NFS 启动脚本启动显示结果如下：

```
[root@localhost CentOS]# service nfs start
启动 NFS 服务:                                            [确定]
关掉 NFS 配额:                                            [确定]
启动 NFS 守护进程:                                        [确定]
启动 NFS mountd:                                          [确定]
```

由以上结果可知，NFS 已经启动成功，NFS 脚本自动启动 rpc.mount 和 rpc.nfsd 两个守护进程。我们可以通过查看进程来检测 NFS 是否启动成功：

```
ps -eaf | grep nfsd
```

显示结果如下所示：

```
[root@localhost CentOS]# ps -eaf | grep nfsd
root     20050     9  0 11:53 ?        00:00:00 [nfsd4]
root     20052     1  0 11:53 ?        00:00:00 [nfsd]
root     20053     1  0 11:53 ?        00:00:00 [nfsd]
root     20054     1  0 11:53 ?        00:00:00 [nfsd]
root     20055     1  0 11:53 ?        00:00:00 [nfsd]
root     20056     1  0 11:53 ?        00:00:00 [nfsd]
root     20057     1  0 11:53 ?        00:00:00 [nfsd]
root     20058     1  0 11:53 ?        00:00:00 [nfsd]
root     20059     1  0 11:53 ?        00:00:00 [nfsd]
root     23022 19241  0 13:15 pts/1    00:00:00 grep nfsd
```

由上面结果可知，NFS 默认会在后台启动 8 个守护进程，当 NFS 检测到客户端连接时，将由 8 个进程中的某个进程接管以提高访问速度与效率。

5.2.3　停止 NFS 服务

关闭 NFS 服务可以使用如下命令：

```
service nfs stop
```

显示结果如下：

```
[root@localhost CentOS]# service nfs stop
关闭 NFS mountd:                                        [确定]
关闭 NFS 守护进程:                                      [确定]
关闭 NFS quotas:                                        [确定]
关闭 NFS 服务:                                          [确定]
```

通过查看进程以确定 NFS 是否已经停止使用：

```
ps -eaf | grep nfsd
```

显示结果如下所示：

```
[root@localhost CentOS]# ps -eaf | grep nfsd
root     24451 19241  0 13:37 pts/1    00:00:00 grep nfsd
```

说明 NFS 守护进程已经停止。

注意： 停止 NFS 服务时，一般 portmap 守护进程不必停止，让其在后台运行即可。

5.2.4 设置 NFS 服务器开机自启动

在字符界面下，可以使用如下命令实现 NFS 服务开机自启动：

```
# chkconfig --level 345 portmap on
# chkconfig --level 345 nfs on
```

另外，也可以使用 ntsysv 命令这个 SysV 风格的 runlevel 配置工具来对开机启动项进行配置。

5.2.5 使用图形化方式设置 NFS 服务

CentOS 5 的 GNOME 图形界面中选择“系统”→“管理”→“服务器设置”→“服务”，打开“服务配置”窗口，如图 5-2 所示。在该窗口中选中 nfs 服务，然后单击“开始”、“停止”、“重启”等按钮即可实现对 nfs 服务的启动、停止和重启操作。在窗口中选中 nfs 复选框，还可实现 nfs 服务的开机自动运行。



图 5-2 CentOS 5 中的“服务配置”窗口

5.3　NFS 服务器的配置

5.3.1　NFS 服务器配置过程

NFS 服务器的配置过程如下：

(1) 编辑 NFS 配置文件/etc/exports。

(2) 重启 portmap 守护进程。

(3) 启动 NFS 服务，并检测服务是否启动成功。

(4) 根据需要设置 NFS 服务开机自动运行。

(5) 客户端挂载 NFS 服务器共享目录并正常使用。

5.3.2　NFS 配置文件

NFS 的配置文件为/etc/exports。NFS 服务安装完成后，配置文件并不存在，需要管理员手工创建。另外也可以通过使用 system-config-nfs 图形化工具来添加其配置信息。

在 exports 文件中，可以定义 NFS 系统的输出目录(即共享目录)、访问权限和允许访问的主机等参数。该文件默认为空，没有配置输出任何共享目录，这是基于安全性的考虑，这样即使系统启动 NFS 服务也不会输出任何共享资源。

exports 文件中每一行提供了一个共享目录的设置，其命令格式为：

＜输出目录＞ ［客户端 1(选项 1，选项 2，…)］ ［客户端 2(选项 1，选项 2，…)］

其中，除输出目录是必选参数外，其他参数都是可选的。格式中的输出目录和客户端之间、客户端与客户端之间都使用空格分隔，但是客户端和选项之间不能有空格。

输出目录是指 NFS 系统中需要共享给客户端使用的目录。

客户端是指网络中可以访问这个 NFS 输出目录的计算机。客户端的指定非常灵活，可以是单个主机的 IP 地址或域名，也可以是某个子网或域中的主机等。客户端常用的指定方式如表 5-1 所示。

表 5-1　客户端常用主机表示办法

客 户 端	说　明
192.168.0.100	指定 IP 地址的主机
192.168.0.0/24(或 192.168.0.*)	指定子网中的所有主机
pc1.test.edu.cn	指定域名的主机
*.test.edu.cn	指定域中的所有主机
*(或缺省)	所有主机

选项用来设置输出目录的访问权限、用户映射等。exports 文件中的选项比较多，一般可分为以下 3 类。

1) 访问权限选项

用于控制输出目录访问权限的选项。这类选项只有 ro 和 rw 两项，如表 5-2 所示。

表 5-2 访问权限选项

访问权限选项	说 明
ro	设置输出目录只读
rw	设置输出目录可读写

2) 用户映射选项

在默认情况下，当客户端访问 NFS 服务器时，若远程访问的用户是 root 用户，则 NFS 服务器会将它映射成一个本地的匿名用户(该用户账户为 nfsnobody)，并将它所属的用户组也映射成匿名用户组(该用户组账户也为 nfsnobody)，这样有助于提高系统的安全性。用户映射选项可对此进行调整，如表 5-3 所示。

表 5-3 用户映射选项

用户映射选项	说 明
all_squash	将远程访问的所有普通用户及所属用户组都映射为匿名用户或用户组(一般均为 nfsnobody)
no_all_squash	不将远程访问的所有普通用户及所属用户组都映射为匿名用户或用户组(默认设置)
root_squash	将 root 用户及所属用户组都映射为匿名用户或用户组(默认设置)
no_root_squash	不将 root 用户及所属用户组都映射为匿名用户或用户组
anonuid=xxx	将远程访问的所有用户都映射为匿名用户，并指定该匿名用户账户为本地用户账户(UID=xxx)
anongid=xxx	将远程访问的所有用户组都映射为匿名用户组账户，并指定该匿名用户组账户为本地用户组账户(GID=xxx)

3) 其他选项

其他选项比较多，可用于对输出目录进行更全面的控制，如表 5-4 所示。

表 5-4 常用的其他选项

其他选项	说 明
secure	限制客户端只能从小于 1024 的 TCP/IP 端口连接 NFS 服务器(默认设置)
insecure	允许客户端从大于 1024 的 TCP/IP 端口连接 NFS 服务器
sync	将数据同步写入内存缓冲区与磁盘中，虽然这样做效率较低，但可以保证数据的一致性
async	将数据先保存在内存缓冲区中，必要时才写入磁盘
wdelay	检查是否有相关的写操作，如果有则将这些写操作一起执行，这样可提高效率(默认设置)
no_wdelay	若有写操作则立即执行，应与 sync 配合使用
subtree_check	若输出目录是一个子目录，则 NFS 服务器将检查其父目录的权限(默认设置)
no_subtree_check	即使输出目录是一个子目录，NFS 服务器也不检查其父目录的权限，这样做可提高效率

5.3.3　NFS 配置文件示例

下面首先给出 NFS 主配置文件 etc/exports 的一个应用实例，然后对有关设置进行说明。

```
/nfs/public  192.168.0.0/24(rw,async)  *(ro)
/nfs/share 192.168.0.100(rw,sync)
/nfs/root *.test.edu.cn(ro,no_root_squash)
/nfs/users *.edu.cn(rw,insecure,all_squash,sync,no_wdelay)
/mnt/cdrom 192.168.0.*(ro)
```

1) /nfs/public　192.168.0.0/24(rw,async)　*(ro)

输出目录/nfs/public 可供子网 192.168.0.0/24 中的所有客户机进行读写操作，而其他网络中的客户机只能读取该目录的内容。

还有一点需要注意：并非用户使用子网 192.168.0.0/24 中的客户机访问该共享目录时就能真正地写入，还要看该目录对该用户有没有开放 Linux 文件系统权限的写入权限。

如果该用户是普通用户，那么只有该目录对该用户开放了写入权限，该用户才可以在该共享目录下创建子目录及文件，且新建子目录及文件的所有者就是该用户(实际上应该是该用户的 UID)。

如果该用户是 root 用户，由于默认选项中有 root_squash，root 用户会被映射为 nfsnobody，因此只有该共享目录对 nfsnobody 开放了写入权限，该用户才能在共享目录中创建子目录及文件，且所有者将变成 nfsnobody。

2) /nfs/share 192.168.0.100(rw,sync)

输出目录/nfs/share 只供 IP 地址为 192.168.0.100 的客户机进行读写操作。

3) /nfs/root　*.test.edu.cn(ro,no_root_squash)

对于输出目录/nfs/root，test.edu.cn 域中的所有客户机都具有只读权限，并且不将 root 用户映射到匿名用户。

4) /nfs/users　*.edu.cn(rw,insecure,all_squash,sync,no_wdelay)

对于输出目录/nfs/users 来说，edu.cn 域中的所有客户机都具有可读可写的权限，并且将所有用户及所属的用户组都映射为 nfsnobody，数据同步写入磁盘。如果有写入操作则立即执行。

5) /mnt/cdrom　192.168.0.*(ro)

对于输出目录/mnt/cdrom 来说，子网 192.168.16.0/24 中的所有客户机都具有只读的权限。通常用户可以将光驱挂载到该文件夹下，实现 NFS 对于光驱的共享。

5.3.4　NFS 服务器端工具

NFS 服务器运行期间，管理员可以使用 NFS 服务器端工具检测服务器的运行情况。如 NFS 的运行状态、RPC 情况、NFS 服务器输出的共享目录情况等。

1. nfsstat 命令

nfsstat 命令用于查看 nfs 的运行状态，对于调试 NFS 的运行有很大帮助。通过该命令能够看到 NFS 服务器和客户端的 RPC 状态，同时可以看到 NFS 服务器和客户端的 NFS

文件更改状态，如更改文件权限、创建文件、创建目录、读文件、写文件等。主要参数如下：

- -s 参数：显示服务器的状态。
- -c 参数：显示客户端的状态。
- -n 参数：显示服务器和客户端的 NFS 状态。
- -r 参数：仅显示服务器和客户端的 RPC 状态。

控制台下直接输入 nfsstat 命令，若 NFS 服务器或客户端有活动连接，则显示结果如下所示：

```
# nfsstat
Server rpc stats:
calls      badcalls   badauth    badclnt    xdrcall
33         0          0          0          0

Server nfs v3:
null        getattr     setattr     lookup      access      readlink
6       19% 7       22% 0        0% 6       19% 3        9% 0        0%
read        write       create      mkdir       symlink     mknod
0        0% 0        0% 2        6% 0        0% 0        0% 0        0%
remove      rmdir       rename      link        readdir     readdirplus
0        0% 0        0% 0        0% 0        0% 0        0% 1        3%
fsstat      fsinfo      pathconf    commit
0        0% 6       19% 0        0% 0        0%

Client rpc stats:
calls      retrans    authrefrsh
24         0          0

Client nfs v3:
null        getattr     setattr     lookup      access      readlink
0        0% 7       30% 0        0% 6       26% 3       13% 0        0%
read        write       create      mkdir       symlink     mknod
0        0% 0        0% 2        8% 0        0% 0        0% 0        0%
remove      rmdir       rename      link        readdir     readdirplus
0        0% 0        0% 0        0% 0        0% 0        0% 1        4%
fsstat      fsinfo      pathconf    commit
0        0% 4       17% 0        0% 0        0%
```

2. exportfs 命令

exportfs 命令用于输出 NFS 服务器的所有共享目录情况。包括服务器共享了哪些目录，允许哪些 IP 地址访问等。Exports 命令还能实现动态地对 NFS 的配置文件/etc/exports 文件进行加载，使之生效。主要参数如下：

- -a 参数：输出在/etc/exports 文件中设置的所有共享目录。
- -r 参数：重新读取/etc/exports 文件中的共享设置，并使之立即生效，而无需重新启动 NFS 服务。
- -u 参数：停止输入共享的某一目录。
- -v 参数：在输出目录时，将详细的信息显示出来。单独使用该参数则显示/etc/exports 配置文件中所有共享目录的详细列表。

下面让我们具体看一下 exportfs 命令的使用方法：

(1) 输入 exportfs 命令查看共享目录：

```
# exportfs -v
/nfs/public
192.168.0.0/24(rw,async,wdelay,root_squash,no_subtree_check,anonuid=6553
4,anongid=65534)
```

(2) 编辑/etc/exports 文件，并添加内容如下：

```
/nfs/share 192.168.0.104(rw,sync)
/nfs/root *.test.edu.cn(ro,no_root_squash)
/nfs/users *.edu.cn(rw,insecure,all_squash,sync,no_wdelay)
/mnt/cdrom 192.168.0.*(ro)
```

(3) 利用 exportfs 命令重新加载配置文件：

```
# exportfs -arv
exporting 192.168.0.0/24:/nfs/public
exporting 192.168.0.104:/nfs/share
exporting *.edu.cn:/nfs/users
exporting 192.168.0.*:/mnt/cdrom
exporting *.test.edu.cn:/nfs/root
exporting *:/nfs/public
```

(4) 在本地 IP 地址为 192.168.0.104 的服务器上挂载/nfs/share 至/nfs 目录：

```
# mount 192.168.0.104:/nfs/public /home/a
# mount 192.168.0.104:/nfs/share /mnt
```

(5) 使用 mount 命令显示挂载情况：

```
# mount
/dev/mapper/VolGroup00-LogVol00 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda1 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
none on /proc/fs/vmblock/mountPoint type vmblock (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/hdc on /media/CentOS_5.5_Final type iso9660
(ro,noexec,nosuid,nodev,uid=0)
nfsd on /proc/fs/nfsd type nfsd (rw)
192.168.0.104:/nfs/share on /mnt type nfs (rw,addr=192.168.0.104)
192.168.0.104:/nfs/public on /home/a type nfs (rw,addr=192.168.0.104)
```

由最后两行可知，服务器目录已经成功挂载至客户端的目录中，且具有了相应的权限。

5.4　NFS 客户端的配置

通过上述方法配置好 NFS 服务器后，需要将 NFS 服务器上共享的目录挂载到本地来使用。此时可以使用 mount 命令挂载，也可以将 NFS 共享目录添加到/etc/fstab 配置文件中，计算机每次开机自动挂载。

5.4.1 使用 showmount 查看 NFS 服务器共享目录

当客户端用户需要了解 NFS 服务器有哪些共享目录时，可以使用 showmount 命令进行查看。showmount 命令的参数如下：

- -a：显示指定的 NFS 服务器的所有客户端主机及其连接的目录。
- -e：导出指定的 NFS 服务器的所有共享目录。没有指定的话将显示本地 NFS 服务器。
- -d：显示指定 NFS 服务器上所有输出的共享目录。

下面我们看一下 showmount 的具体用法。

(1) 显示 IP 地址为 192.168.0.104 的 NFS 服务器上所有的共享目录：

```
# showmount -d 192.168.0.104
```

显示结果如下：

```
# showmount -d 192.168.0.104
Directories on 192.168.0.104:
/nfs/public
/nfs/share
```

说明服务器已经共享/nfs/public/、/nfs/share 目录。

(2) 在客户端上显示 192.168.0.104 服务器的所有 NFS 客户端及连接目录：

```
# showmount -a 192.168.0.104
```

显示结果如下：

```
# showmount -a 192.168.0.104
All mount points on 192.168.0.104:
192.168.0.103:/nfs/public
192.168.0.104:/nfs/public
192.168.0.104:/nfs/share
```

说明 192.168.0.104 服务器上共享了/nfs/public、/nfs/share 目录。同时 192.168.0.103 已经连接到了服务器上的/nfs/public 目录；192.168.0.104 已经连接到/nfs/public、/nfs/share 两个目录。

注意： SELinux 及 Linux 防火墙将使客户端无法访问服务器，需要在服务器端进行如下配置：

```
# setsebool -P portmap_disable_trans=1
# setsebool -P nfs_export_all_rw=1
# setsebool -P nfs_export_all_ro=1
# setsebool -P use_nfs_home_dirs 1
# service iptables stop
```

5.4.2 挂载 NFS 服务器目录

在 CentOS5 中挂载 NFS 服务器目录可以通过使用 mount 命令实现。命令格式如下：

```
mount -t NFS 服务器目录 本地挂载目录
```

目前，mount 命令能够直接对 NFS 进行支持，所以也可以直接执行如下命令，完成 NFS 服务器的挂载：

```
mount  服务器目录 本地挂载目录
```

例如假设客户端需要将 IP 地址为 192.168.0.104 的 NSF 服务器上的 /nfs/share 目录挂载到本地的/usr/local/nfs 目录下，使用以下命令：

首先创建/usr/local/nfs 目录：

```
#mkdir /usr/local/nfs
```

然后使用 mount 命令挂载服务器目录：

```
#mount 192.168.0.104:/nfs/share /usr/local/nfs
```

命令执行完成，即可实现服务器目录的加载。

如果要卸载刚刚挂载的 NSF 服务器目录，执行以下命令：

```
umount /usr/local/nfs
```

5.4.3　设置开机自动挂载 NFS

在上面的例子中，使用 mount 命令挂载的 NFS 目录在重新启动计算机后是不能实现自动挂载的。在 CentOS 5 中，开机自动加载文件系统是在/etc/fstab 中定义的，NFS 文件系统的自动挂载也可以在/etc/fstab 中进行设置。

首先使用文本编辑器打开/etc/fstab 文件，执行如下命令：

```
#vi /etc/fstab
```

在文件末端加入：

```
192.168.0.104:/nfs/share /usr/local/nfs  nfs defaults 0 0
```

执行重新加载 fstab 文件中定义的文件系统使之生效：

```
#mount -a
```

通过上述设置，每次客户端 Linux 开机都能实现对 NSF 服务器的自动加载。

5.5　图形界面配置 NFS 服务器

在 Linux 的 X Windows 下，可以通过图形界面的 NFS 的配置工具完成对 NFS 服务器的配置工作。图形界面配置 NFS 服务器的软件包名称为 system-config-nfs-*.rpm。可以采用以下步骤安装配置服务器。

(1) 查询是否已经安装，若未安装则安装 system-config-nfs-*.rpm 软件包(*为该软件版本号)：

```
# rpm -qa | grep system-config-nfs
```

若无输出，则说明软件包没有安装。

```
# cd /media/CentOS*/Cent*/
# rpm -ivh system-config-nfs-*.rpm
```

(2) 配置 NFS 服务器。在命令行输入 system-config-nfs 或者在 X Windows 中选择“系统”→“管理”→“服务器设置”→NFS，打开如图 5-3 所示的窗口。

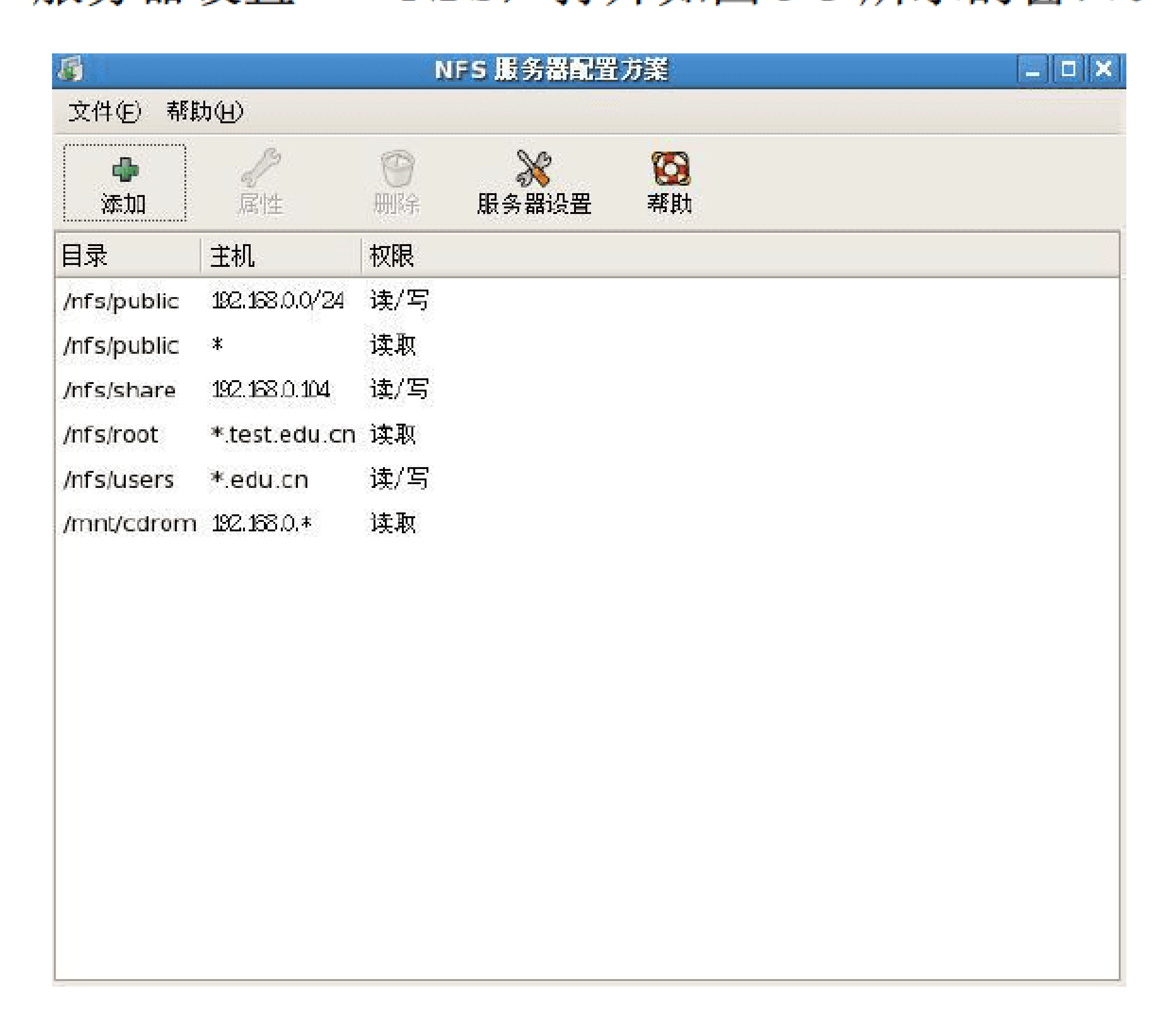


图 5-3　服务器配置方案

(3) 在 NFS 服务器配置管理方案界面中，可以对 NFS 资源进行添加、修改、删除，并可对服务器端口参数进行设置。单击“添加”按钮，向服务器添加一条 NFS 共享资源。在打开的如图 5-4 所示对话框的“基本”选项卡中，在“目录”文本框中输入需 NFS 共享的目录，在“主机”文本框中输入允许哪些主机或者网段访问 NFS 共享。

(4) 在如图 5-5 所示的界面中设置 NFS 共享目录的选项内容。其选项所对应的配置文件选项如表 5-5 所示。




图 5-4　添加 NFS 共享基本

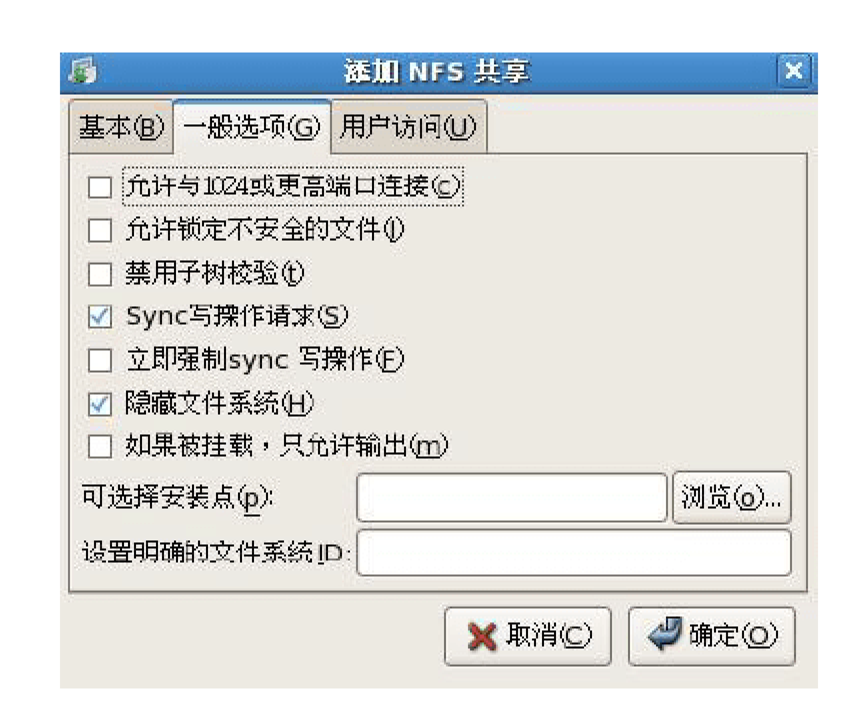


图 5-5　添加 NFS 共享一般选项

表 5-5　一般选项与配置文件选项对应表

复 选 框	配置文件选项
允许与 1024 或更高端口连接	insecure
允许锁定不安全的文件	insecure_lock

续表

复 选 框	配置文件选项
禁用子树校验	no_subtree_check
Sync 写操作请求	sync
立即强制 sync 写操作	no_wdelay

(5) 在如图 5-6 的用户访问选项卡中设置 NFS 共享的用户权限控制。与配置文件选项的对应关系如表 5-6 所示。

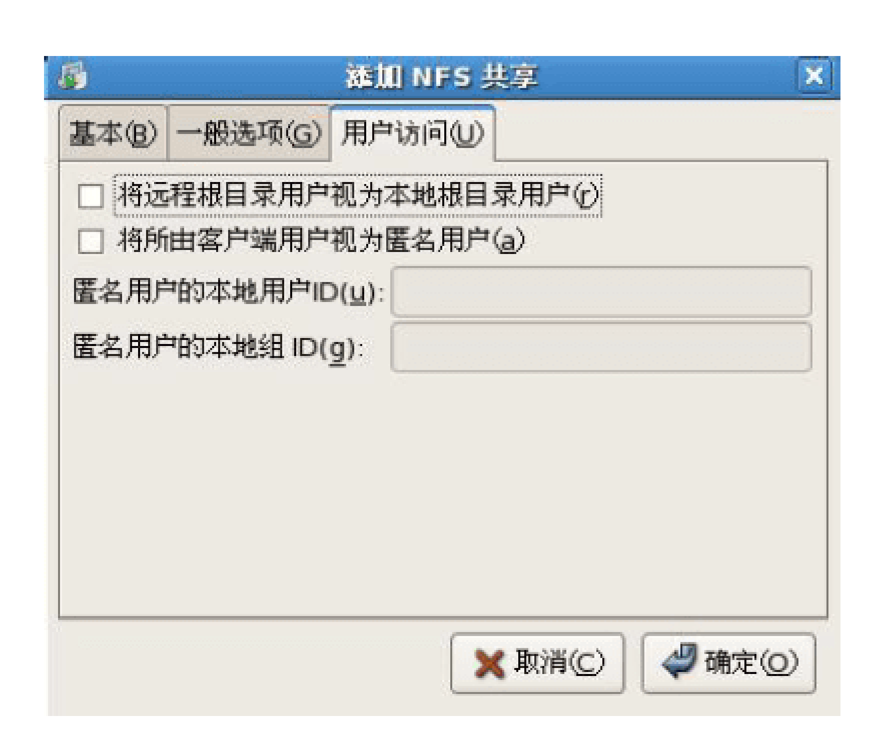


图 5-6　添加 NFS 共享用户访问

表 5-6　用户访问与配置文件选项对应表

用户访问选项	配置文件选项
将远程根目录用户视为本地根目录用户	no_root_squash
将所有客户端用户视为匿名用户	all_squash
匿名用户的本地用户 ID	anonuid=xxx
匿名用户的本地组	anongid=xxx

5.6　NFS 服务的配置案例

【例 5-1】假设 Linux 服务器 IP 地址为 192.168.0.105。

通过 NFS 共享/nfs/tmp 目录给 192.168.0.0/24 网段中的所有计算机，权限为可读写。

共享/nfs/nfs 目录给所有计算机可读权限。

共享/nfs/upload 目录作为 192.168.0.0/24 网段计算机的上传目录，其中/home/upload 的用户及所属组的名字为 nfs-upload，UID 与 GID 均为/etc/fstab。

将/home/alice 这个目录共享给 192.168.0.104 这台主机的 alice 账户来使用，也就是说 alice 在 192.168.0.104 及 192.168.0.105 均有账号，且账号均为 alice，所以开发/home/alice 目录给 alice 使用它的默认目录。

下面将按照上述要求完成 NFS 服务器的搭建。

5.6.1 服务器配置

(1) NFS 修改本机 IP 地址：

```
# ifconfig eth0 192.168.0.105
```

(2) 配置/etc/exports 配置文件，修改内容如下：

```
# vi /etc/exports

/nfs/tmp  192.168.0.*(rw,async)
/nfs/nfs *(ro,all_squash)
/nfs/upload 192.168.0.*(rw,all_squash,anonuid=100,anongid=100)
/home/alice 192.168.0.104(rw)
```

(3) 建立目录及权限：

```
# cd /
# mkdir nfs
# cd nfs
# mkdir tmp nfs upload
# chmod 777 -R tmp
# chmod 755 -R upload

# groupadd -g 100 nfs-upload
# useradd -g 100 -u 100 -M nfs-upload
# chown -R nfs-upload:nfs-upload /nfs/upload
```

(4) 添加用户 alice：

```
# useradd alice
```

(5) 启动 portmap 与 nfs 服务：

```
# service portmap start
启动 portmap:                                              [确定]

# service nfs start
启动 NFS 服务:                                             [确定]
关掉 NFS 配额:                                             [确定]
启动 NFS 守护进程:                                         [确定]
启动 NFS mountd:                                            [确定]
```

(6) 打开 SELinux 并关闭防火墙：

```
# setsebool -P portmap_disable_trans=1
# setsebool -P nfs_export_all_rw=1
# setsebool -P nfs_export_all_r0=1
# setsebool -P use_nfs_home_dirs 1
# service iptables stop
```

5.6.2 客户端配置

(1) NFS 修改本机 IP 地址：

```
# ifconfig eth0 192.168.0.104
```

(2) 启动 protmap 服务：

```
# service portmap start
```

(3) 创建 alice 用户

```
useradd alice
```

(4) 查询服务器上的挂载点：

```
# showmount -e 192.168.0.105
Export list for 192.168.0.105:
/nfs/nfs    *
/nfs/tmp    192.168.0.*
/nfs/upload 192.168.0.*
/home/alice 192.168.0.104
```

(5) 建立挂载点目录：

```
# mkdir /mnt/tmp /mnt/nfs /mnt/upload /mnt/alice
```

(6) 挂载 NFS 服务目录：

```
# mount 192.168.0.105:/nfs/tmp /mnt/tmp
[root@bogon mnt]# mount 192.168.0.105:/nfs/nfs /mnt/nfs
[root@bogon mnt]# mount 192.168.0.105:/nfs/upload /mnt/upload
[root@bogon mnt]# mount 192.168.0.105:/home/alice /mnt/alice
```

至此，本案例配置完成。

5.6.3　客户端测试

(1) 使用 root 用户登录客户端，查看 NFS 挂载情况，显示结果如下所示：

```
# mount
/dev/mapper/VolGroup00-LogVol00 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda1 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
none on /proc/fs/vmblock/mountPoint type vmblock (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/hdc on /media/CentOS_5.5_Final type iso9660
(ro,noexec,nosuid,nodev,uid=0)
192.168.0.105:/nfs/tmp on /mnt/tmp type nfs (rw,addr=192.168.0.105)
192.168.0.105:/nfs/nfs on /mnt/nfs type nfs (rw,addr=192.168.0.105)
192.168.0.105:/nfs/upload on /mnt/upload type nfs (rw,addr=192.168.0.105)
192.168.0.105:/home/alice on /mnt/alice type nfs (rw,addr=192.168.0.105)
```

(2) 测试/mnt/tmp 及/mnt/upload 目录是否可写：

```
# touch  /mnt/tmp/1.txt
# touch  /mnt/upload/1.txt
```

执行上述命令没有提示错误，说明/mnt/tmp 及/mnt/upload 两个目录加载正常，具有可写权限。

(3) 测试/nfs 是否为可读权限：

```
# touch  /mnt/nfs/1.txt
touch: 无法触碰 “/mnt/nfs/1.txt”: 只读文件系统
```

(4) 测试/mnt/alice 目录 root 用户是否具有权限：

```
# cd /mnt/alice
bash: cd: /mnt/alice: 权限不够
```

由此可知，root 用户没有 alice 目录的相应权限，切换用户至 alice，再次测试：

```
# su alice
$ cd /mnt/alice
$ touch a.txt
$ ls -l
总计 32
drwx------ 3 alice  alice 4096 03-29 04:05 alice
drwxr-xr-x 2 root  root 4096 03-29 02:23 nfs
drwxrwxrwx 2 root  root 4096 03-29 03:39 tmp
drwxr-xr-x 2   100  100 4096 03-29 03:39 upload
```

由上述命令可知，alice 用户可以对/mnt/alice 目录进行读写。但必须注意的是，若要实现上述功能，服务器的 alice 用户必须与客户端的 alice 用户必须有相同的 UID 与组 GID；如不相同，用 usermod 命令修改一致即可。

注意： 如果客户端无法挂载 NFS，应从以下几个方面排查原因：

① 用户或客户端身份权限不符。

② 服务器或客户端 portmap 服务没有启动。

③ 被 SELinux 或者防火墙拦截。

5.7 本章小结

NFS 是分布式计算机系统的一个组成部分，可实现在异构网络上共享和装配远程文件系统。本章首先介绍了 NFS 服务器的工作原理；之后介绍了安装运行 NFS 服务器的方法以及 NFS 服务器与客户端具体的配置步骤；同时本章也介绍了图形化配置 NFS 服务器的方法与过程；最后用一个配置案例来总结 NFS 服务器、客户端的配置方法，方便读者自行学习和研究 NFS 服务器。

5.8 课后练习

1. 填空题

(1) NFS 是________英文的简称，请中文含义是___________。

(2) NFS 服务的三个守护进程分别是：________、________、__________。

(3) NFS 主要用于________和__________ 操作系统共享文件时使用。

(4) 显示 192.168.1.103 服务器上所有的客户端主机及连接目录应该使用__________命令。

(5) 设置 NFS 服务在级别 345 时开机自启动的两个命令是____________和____________。

2. 选择题

(1) 在 bash 中，export 命令的作用是(　　)。

A. 在子 shell 中运行条命令

B. 使在子 shell 中可以使用命令历史记录

C. 为其他应用程序设置环境变量

D. 提供 NFS 分区给网络中的其他系统使用

(2) 下面(　　)文件包含了供 NFS daemon 使用的目录列表。

A. /etc/nfs　　B. /etc/nfs.conf

C. /etc/exports　　D. /etc/netdir

(3) NFS 工作站要 mount 远程 NFS 服务器上的一个目录的时候，以下(　　)是服务器端必需的。

A. portmap 必须启动

B. NFS 服务必须启动

C. 共享目录必须加装在/etc/exports 文件里

D. 以上全部都需要

(4) (　　)命令可以完成加装 NFS 服务器 www.edu.cn 的/home/nfs 共享目录到本机/home2。

A. mount –t nfs www.edu.cn:/home/nfs /home2

B. mount –t –s nfs www.edu.cn./home/nfs /home2

C. nfsmount www.edu.cn:/home/nfs /home2

D. nfsmount –s www.edu.cn/home/nfs /home2

(5) (　　)命令用来查看 NFS 磁盘资源被其他系统使用情况。

A. share　　B. mount　　C. export　　D. exports

(6) 以下 NFS 系统中关于用户 ID 映射正确的描述是(　　)。

A. 服务器上的 root 用户默认值和客户端的一样

B. root 默认被映射到 nfsnobody 用户

C. root 默认不被映射到 nfsnobody 用户

D. 默认情况下，anomuid 不需要密码。

(7) 在你公司有 10 台 Linux Server，你想用 NFS 在 Linux Server 之间共享文件，应该修改的文件是(　　)。

A. /etc/exports　　B. /etc/crontab

C. /etc/named.conf　　D. /etc/fstab

(8) 查勘 NFS 服务器 192.168.0.1 中共享目录的命令是(　　)。

A. show –e 192.168.0.1　　B. show //192.168.0.1

C. showmount –e 192.168.0.1　　D. showmount –l 192.168.0.1

(9) 装载 NFS 服务器 192.168.12.1 的共享目录/tmp 到本地目录/mnt/share 的命令是

(　　)。

A. mount 192.168.12.1/tmp　/mnt/share

B. mount –t nfs 192.168.12.1/tmp　/mnt/share

C. mount –t nfs 192.168.12.1:/tmp　/mnt/share

D. mount –t nfs //192.168.12.1/tmp　/mnt/share

(10) 需要在 NFS 客户端设置共享目录开机自启动，应修改(　　)配置文件。

A. /etc/exports　　B. /etc/fstab

C. /etc/exports.conf　　D. /etc/fstab.conf

3. 简答题

(1) 什么是 NFS？挂载 NFS 的方法是什么？

(2) 简述 NFS 三个守护进程的作用。

第6章

DNS服务器安装与配置

域名服务(DNS)是网络中最重要的网络服务之一，它是一个分布式数据库组织成域层次结构的计算机和网络服务命名系统。DNS 服务可以方便地让难以记忆的 IP 地址取代为便于人们记忆的域名，从而使人们能用简单好记的域名来代替 IP 地址访问网络。本章将详细介绍 DNS 服务的基本概念、工作原理、服务器架设及使用方法。

6.1 DNS 服务概述

DNS(Domain Name Server，域名服务器)是为了让用户方便访问 Internet 而采用的一种分布式域名与 IP 地址之间的映射、查询和管理的方法。在现有的 Internet 网络结构中，计算机之间的通信只能通过其唯一标识 IP 地址才能进行。但是 IP 地址难以记忆，所以就采用“域名”的方式代替这些数字。不过最终还是必须有一种机制将域名转换成对应的 IP 地址才能访问主机，因此需要一种将主机转换为 IP 地址的机制。

6.1.1 域名的解析方法

早期域名与 IP 地址的对应关系表是记录在每台计算机中的 hosts 文件中，当网络内计算机数目不多时，hosts 文件通过定期更新可以满足计算机通信要求。但是当网络规模逐渐扩大，计算机数目呈现出指数增长的趋势，使用 hosts 文件记录域名与 IP 地址映射的方式变得难以维持，所以逐渐发展出了 DNS 服务器。

1. hosts 表

hosts 表是一个简单的文本文件，记录了主机名与 IP 地址的映射关系，计算机通过在该文件中搜索相应的条目来匹配主机名和 IP 地址。在 Linux 操作系统中，hosts 文件位于 /etc/hosts；在 Windows 操作系统中，hosts 文件位于 C:\Windows\System32\drivers\etc\hosts。hosts 文件中每一行就是一个条目，包含一个 IP 地址与该 IP 地址相关联的主机名称(域名)。网络中加入、删除或者重命名主机后，计算机管理员都要对 hosts 文件进行更新以便计算机能通过域名与其他计算机进行通信。

20 世纪 90 年代之后，Internet 网络中计算机数量爆炸式增长，通过一个中心授权机构为所有的 Internet 主机管理一个 hosts 文件变得不再可行。hosts 文件随着主机数量的增多而变得庞大，而将其及时地更新到每一个联网主机变得异常困难。

虽然 hosts 表不再使用，但是大部分操作系统都保留了它的功能。主机对域名的解析的时候首先查询 hosts 表文件中是否有对应记录，如没有再使用 DNS 功能进行域名解析。通过用户手工添加记录，hosts 表还可以用来屏蔽网络中的有害或者病毒网站。

2. NIS 系统

将主机转换为 IP 地址的另一种方案是 NIS(Network Information System，网络信息系统)。NIS 是由 Sun Microsystem 开发的一种命名系统。NIS 将主机表替换成主机数据库，客户机可以从数据库中得到所需要的主机信息。NIS 将所有的数据保存在中央主机上，再由中央主机将所有数据分配给用户，所以其主机名与 IP 地址的解析效率较低。NIS 一般只在中小型网络中应用。

3. DNS 系统

DNS 系统是一种新的主机名称和 IP 地址转换机制，它使用一种分层的分布式数据库来处理 Internet 上众多主机的 IP 地址转换。也就是说，网络中没有存放全部 Internet 主机

信息的中心数据库，这些信息分布在一个层次结构中的若干台域名服务器上。DNS 是基于客户端/服务器模式设计的。本质上，整个域名系统以一个大的分布式数据库方式工作。具有 Internet 连接的企业网络都可以有一个域名服务器，每个域名服务器包含有指向其他域名服务器的信息，结果是这些服务器形成了一个大的协调工作的域名数据库。

6.1.2　DNS 组成

每当一个应用需要将域名翻译成为 IP 地址时，这个应用便成为域名系统的一个客户。这个客户将待翻译的域名放在一个 DNS 请求信息中，并将这个请求发给域名空间中的 DNS 服务器。服务器从请求中取出域名，将它翻译为对应的 IP 地址，然后在一个回答信息中将结果返回给应用。如果接到请求的 DNS 服务器自己不能把域名翻译为 IP 地址，将向其他 DNS 服务器查询。整个 DNS 域名系统由以下 3 个部分组成。

1．DNS 域名空间

指定用于组织名称的域的层次结构，它如同一棵倒立的树，层次结构非常清晰，如图 6-1 所示。根域位于顶部，紧接着在根域的下面是几个顶级域，每个顶级域又可以进一步划分为不同的二级域，二级域再划分出子域，子域下面可以是主机也可以是再划分的子域，直到最后的主机。在 Internet 中的域是由 InterNIC 负责管理的，域名的服务则由 DNS 来实现。

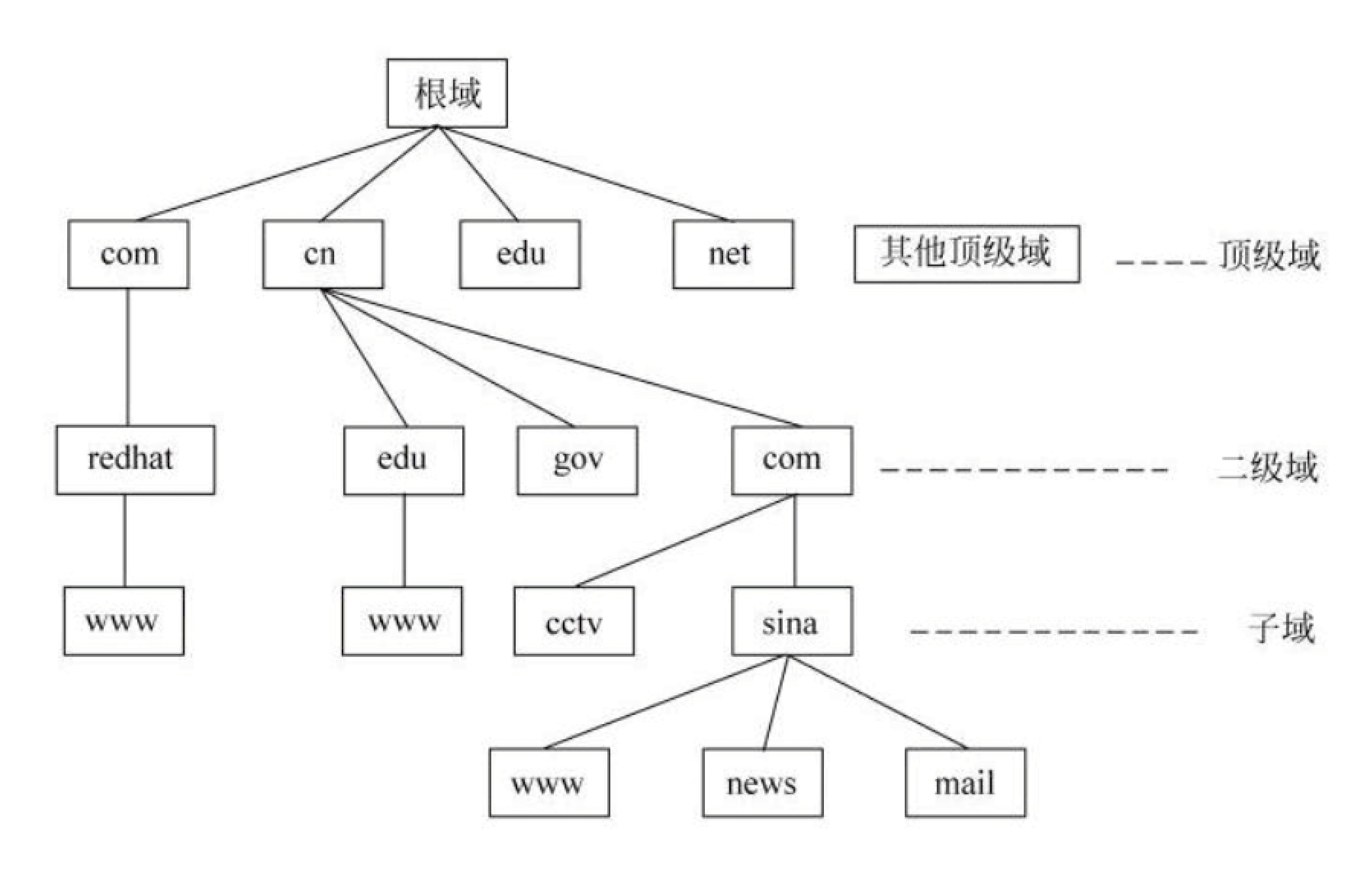


图 6-1　DNS 域名空间

2．DNS 服务器

DNS 服务器是保持和维护域名空间中数据的程序。由于域名服务是分布式的，每一个 DNS 服务器含有一个域名空间自己的完整信息，其控制范围称为区(Zone)。对于本区内的请求由负责本区的 DNS 服务器解释，对于其他区的请求将由本区的 DNS 服务器与负责该区的相应服务器联系。

3. **解析器**

解析器是简单的程序或子程序，它从服务器中提取信息以响应对域名空间中主机的查询，用于 DNS 客户端。

6.1.3 DNS 查询过程

当客户端程序要通过一个主机名称来访问网络中的一台主机时，它首先要得到这个主机名称所对应的 IP 地址，因为 IP 数据报中允许放置的是目地主机的 IP 地址，而不是主机名称。可以从本机的 hosts 文件中得到主机名称所对应的 IP 地址，但如果 hosts 文件不能解析该主机名称时，只能通过向客户机所设定 DNS 服务器进行查询了。

可以以不同的方式对 DNS 查询进行解析：第 1 种是本地解析，就是客户端可以使用缓存信息就地应答，这些缓存信息是通过以前的查询获得的；第 2 种是直接解析，就是直接由所设定的 DNS 服务器解析，使用的是该 DNS 服务器的资源记录缓存或者其权威回答(如果所查询的域名是该服务器管辖的)；第 3 种是递归查询，即设定的 DNS 服务器代表客户端向其他 DNS 服务器查询，以便完全解析该名称，并将结果返回至客户端。第 4 种是迭代查询，即设定的 DNS 服务器向客户端返回一个可以解析该域名的其他 DNS 服务器，客户端再继续向其他 DNS 服务器查询。

1. **本地解析**

客户机平时得到的 DNS 查询记录都保留在 DNS 缓存中，客户机操作系统上都运行着一个 DNS 客户端程序。当其他程序提出 DNS 查询请求时，这个查询请求要传送至 DNS 客户端程序。DNS 客户端程序首先使用本地缓存信息进行解析，如果可以解析所要查询的名称，则 DNS 客户端程序就直接应答该查询，而不需要向 DNS 服务器查询，该 DNS 查询处理过程也就结束了，如图 6-2 所示。

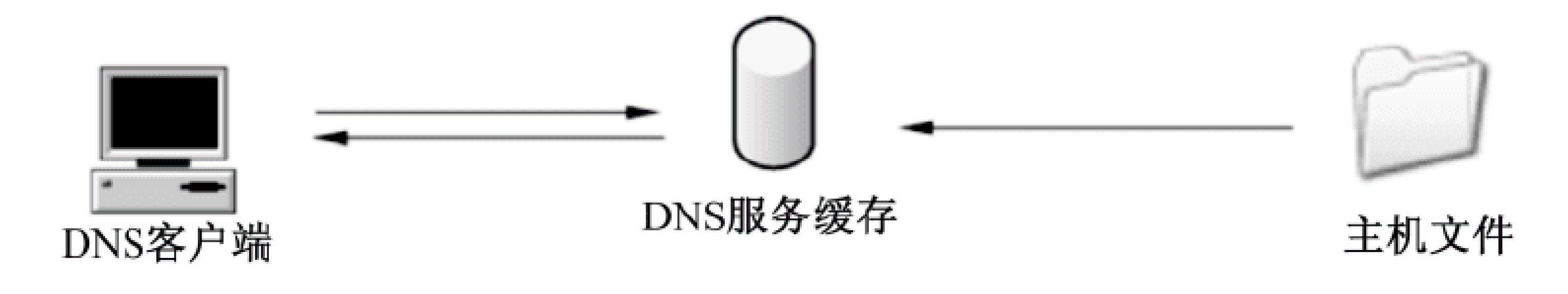


图 6-2　DNS 本地解析

2. **直接解析**

如果 DNS 客户端程序不能从本地 DNS 缓存回答客户机的 DNS 查询，它就向客户机所设定的局部 DNS 服务器发一个查询请求，要求局部 DNS 服务器进行解析。如图 6-3 所示，局部 DNS 服务器得到这个查询请求，首先查看一下所要求查询的域名是不是自己能回答的，如果能回答，则直接给予回答，如果不能回答，再查看自己的 DNS 缓存，如果可以从缓存中解析，则也是直接给予回应。

3. **递归查询**

当局部 DNS 服务器自己不能回答客户机的 DNS 查询时，它就需要向其他 DNS 服务器进行查询。此时有两种方式，图 6-4 所示的是递归查询方式。局部 DNS 服务器自己负

责向其他 DNS 服务器进行查询，一般是先向该域名的根域服务器查询，再由根域名服务器一级级向下查询。最后得到的查询结果返回给局部 DNS 服务器，再由局部 DNS 服务器返回给客户端。

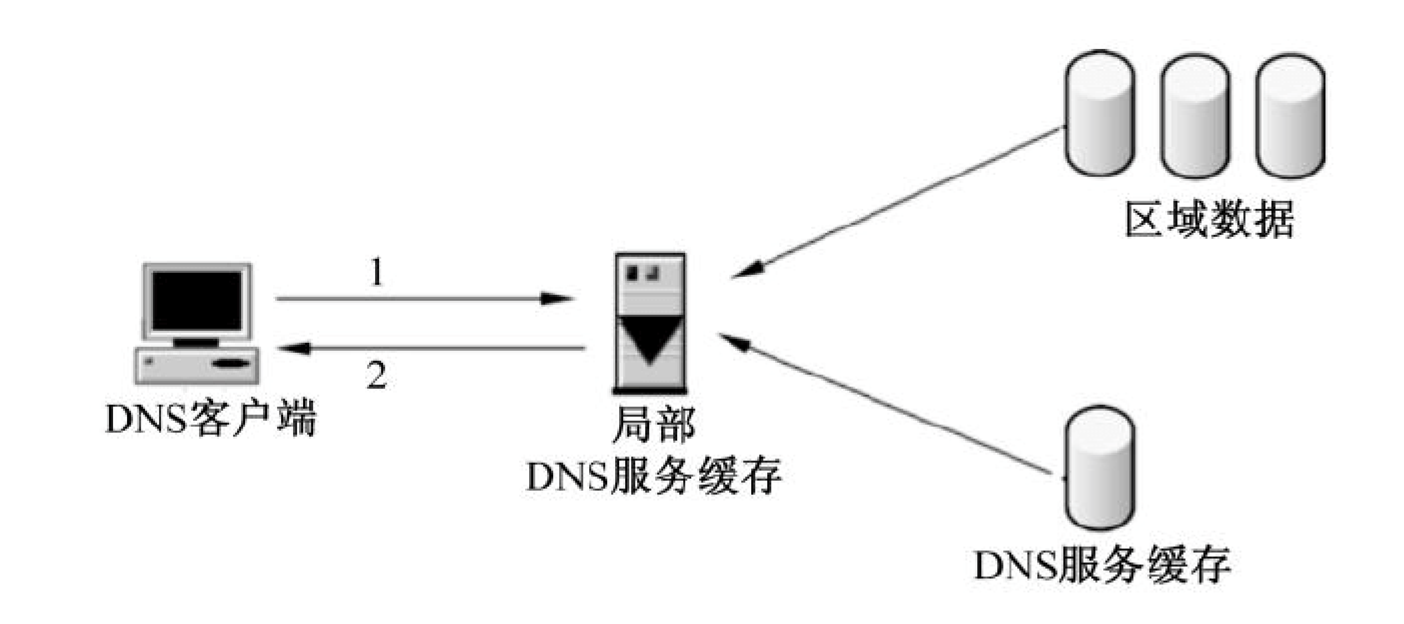


图 6-3　DNS 直接解析

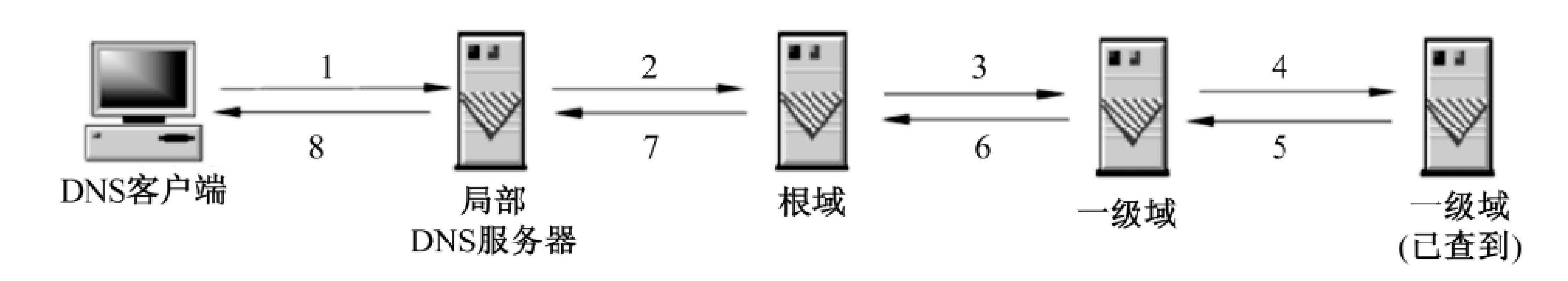


图 6-4　DNS 递归查询

4．迭代查询

当局部 DNS 服务器自己不能回答客户机的 DNS 查询时，也可以通过迭代查询的方式进行解析，如图 6-5 所示。局部 DNS 服务器不是自己向其他 DNS 服务器进行查询，而是把能解析该域名的其他 DNS 服务器的 IP 地址返回给客户端 DNS 程序，客户端 DNS 程序再继续向这些 DNS 服务器进行查询，直到得到查询结果为止。

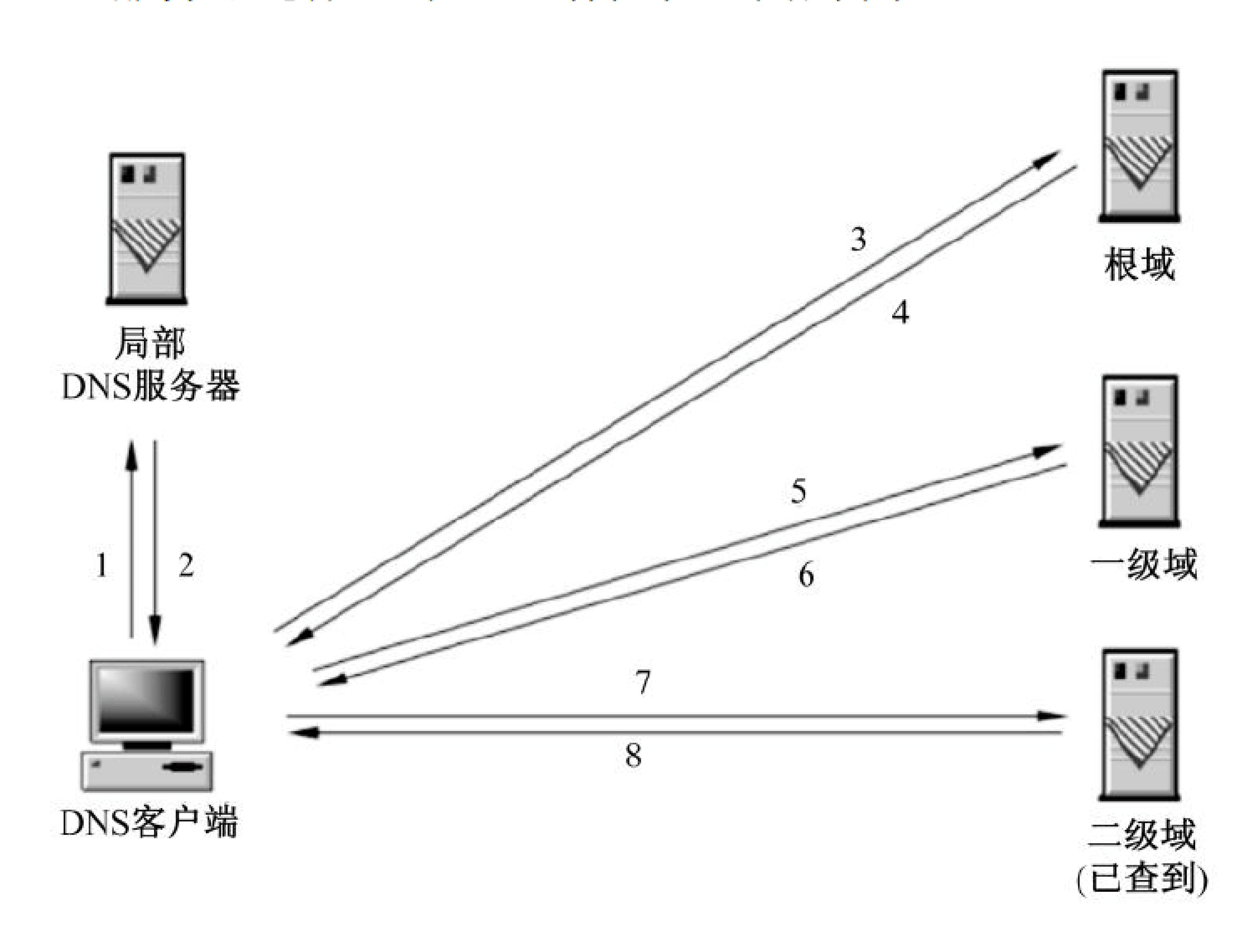


图 6-5　DNS 迭代查询

6.2 BIND 简介

Linux 下架设 DNS 服务器通常是使用 BIND 程序来实现的。BIND 是 Berkeley Internet Name Domain Service 的简写，它是一款实现 DNS 服务器的开源软件。BIND 原本是美国 DARPA 资助加州大学伯利克分校(Berkeley)开设的一个研究生课题，后来经过多年的变化发展，已经成为世界上使用最为广泛的 DNS 服务器软件，目前 Internet 上绝大多数的 DNS 服务器都是用 BIND 来架设的。

BIND 能够运行在当前大多数的操作系统平台之上。目前 BIND 软件由 ISC(Internet Software Consortium，因特网软件联合会)这个非营利性机构负责开发和维护。ISC 的官方网站 http://www.isc.org/，可以下载该软件包的最新版本。

BIND 软件包包括 3 个部分：

- BIND 服务器。这是一个叫做 named 的程序，代表 Name Daemon 的简写。它根据 DNS 协议标准的规定，响应收到的查询。
- BIND 解析库。一个解析器是一个程序，通过发送请求到合适的服务器并且对服务器的响应做出合适的回应，来解析对一个域名的查询。一个解析库是程序组件的集合，可以在开发其他程序时使用，为这些程序提供域名解析的功能。
- 测试服务器的软件工具。主要包括服务器端工具和客户端工具两个部分。

6.3 BIND 服务的安装与运行

6.3.1 BIND 服务安装

在安装 BIND 服务之前，我们应使用下面的命令检查系统是否已经安装了 DNS 服务或查看已经安装了何种版本。

```
# rpm -qa  bind
```

命令执行后没有返回结果，需要对 BIND 服务软件包进行安装。首先将 CentOS 5 的光盘加载，进入到 rpm 包所在目录进行安装：

```
# cd /media/CentOS*/CentOS
# rpm -ivh bind-*.rpm --force
Preparing...                ########################################### [100%]
   1:bind-libs              ########################################### [ 14%]
   2:bind                   ########################################### [ 29%]
   3:bind-utils             ########################################### [ 43%]
   4:bind-chroot            ########################################### [ 57%]
   5:bind-devel             ########################################### [ 71%]
   6:bind-libbind-devel     ########################################### [ 86%]
   7:bind-sdb               ########################################### [100%]
```

下面逐一介绍各个软件包功能：

- bind-devel-9.3.6-4.P1.el5_4.2.rpm 是 BIND 服务开发包。
- bind-9.3.6-4.P1.el5_4.2.rpm 是 BIND 服务的主要软件包。
- bind-utils-9.3.6-4.P1.el5_4.2.rpm 是 BIND 服务常用软件包。

- bind-sdb-9.3.6-4.P1.el5_4.2.rpm 是 BIND 服务数据库后端软件包。
- bind-chroot-9.3.6-4.P1.el5_4.2.rpm 是让 BIND 在 chroot 模式运行的软件包。
- bind-libs-9.3.6-4.P1.el5_4.2.rpm 是 BIND 服务的动态链接库软件包。
- bind-libbind-devel-9.3.6-4.P1.el5_4.2.rpm 是 BIND 服务动态链接库开发软件包。

另外，安装完成 BIND 后，我们应选择安装 caching-nameserver-*.rmp 包，该软件包包含了 BIND 服务的配置模版文件，能够方便对 BIND 配置文件进行配置与修改。

```
# rpm -ivh caching-nameserver-*
Preparing...                ########################################### [100%]
```

注意： bind 软件包安装后，在/usr/share/doc/bind-9.3.6/sample/目录下包含了 bind 服务的各种配置文件，我们也可以直接复制到配置文件工作目录中使用。

6.3.2　BIND 服务运行与停止

BIND 服务的启动脚本文件为/etc/init.d/named 文件。该文件不仅控制 BIND 服务的启动、停止、重启等，同时也能实现动态加载区域数据库文件及查看 BIND 服务的运行状态。通过 service named 命令调用可实现其功能，命令格式如下所示：

```
# service named {start|stop|status|restart|condrestart|reload|probe}
```

1. 启动 BIND 服务

```
# service named start
```

返回结果如下所示，说明 BIND 服务已经启动。

```
# service named start
启动 named：                                               [确定]
```

2. 停止 BIND 服务

```
# service named stop
```

返回结果如下所示，说明 BIND 服务已经停止。

```
# service named stop
停止 named：                                               [确定]
```

3. 重新启动 BIND 服务器

```
# service named restart
停止 named：.                                              [确定]
启动 named：                                               [确定]
```

4. 使用 ps 命令检查 named 进程情况

```
# ps -ef | grep named
named    30253     1  0 08:00 ?        00:00:00 /usr/sbin/named -u named -c
/etc/named.caching-nameserver.conf -t /var/named/chroot
root     30315 23451  0 08:03 pts/4    00:00:00 grep named
```

5. 使用 netstat 检查 named 运行的端口

```
# netstat -nutap | grep named
tcp        0      0 192.168.0.104:53          0.0.0.0:*
LISTEN     30253/named
tcp        0      0 127.0.0.1:953            0.0.0.0:*
LISTEN     30253/named
tcp        0      0 ::1:953          :::*          LISTEN      30253/named
udp        0      0 192.168.0.104:53      0.0.0.0:*           30253/named
```

6. 设置 DHCP 服务器开机自启动

```
chkconfig --level 345 named  on
```

7. 使用图形化方式设置 BIND 服务器

CentOS 5 中自带了一种很好的设置服务器各种服务的工具。选择“系统”→“管理”→“服务器设置”→“服务”命令，打开 CentOS 中的“服务配置”窗口，如图 6-6 所示。在该窗口中选中 dhcpd 服务，然后单击“开始”、“停止”、“重启”等按钮即可实现对 dhcp 服务的启动、停止和重启操作。在窗口中选中服务前的复选框，还可实现 named 服务的开机自动运行。



图 6-6 CentOS 5 中的“服务配置”窗口

6.3.3 rndc 的使用

rndc 是 BIND 安装包提供的一种控制域名服务运行的工具，它可以在不停止 DNS 服务器工作的情况进行数据的更新，使修改后的配置文件生效。在 DNS 生产应用的时候，DNS 服务器是非常繁忙的，任何短时间的服务暂停都会使用户服务正常使用网络。因此，使用 rndc 工具可以使我们方便的对 DNS 服务器进行不间断服务的修改。

rndc 与 DNS 服务器实行连接时，需要通过数字证书进行认证，而不是传统的用户名/密码方式。在当前版本下，rndc 和 named 都只支持 HMAC-MD5 认证算法，在通信两端使用共享密钥。rndc 在连接通道中发送命令时，必须使用经过服务器认可的密钥加密。为了生成双方都认可的密钥，可以使用 rndc-confgen 命令产生密钥和相应的配置，再把这些配

置分别放入 named.conf 和 rndc 的配置文件 rndc.conf 中，具体操作步骤如下所示。

(1) 执行 rndc-confgen 命令，得到密钥和相应的配置。

```
# rndc-confgen
# Start of rndc.conf
key "rndckey" {
        algorithm hmac-md5;
        secret "DAab8G5BimvFjho2509Wag==";
};

options {
        default-key "rndckey";
        default-server 127.0.0.1;
        default-port 953;
};
# End of rndc.conf

# Use with the following in named.conf, adjusting the allow list as
needed:
# key "rndckey" {
#       algorithm hmac-md5;
#       secret "DAab8G5BimvFjho2509Wag==";
# };
#
# controls {
#       inet 127.0.0.1 port 953
#               allow { 127.0.0.1; } keys { "rndckey"; };
# };
# End of named.conf
```

(2) 在/etc 目录下创建 rndc.conf 文件，输入(1)中不带注释的内容：

```
 # vi rndc.key
key "rndckey" {
        algorithm hmac-md5;
        secret "DAab8G5BimvFjho2509Wag==";
};

options {
        default-key "rndckey";
        default-server 127.0.0.1;
        default-port 953;
};
```

(3) 根据(1)中提示，把下列内容放入/etc/named.conf 中：

```
key "rndckey" {
       algorithm hmac-md5;
       secret "DAab8G5BimvFjho2509Wag==";
 };

 controls {
       inet 127.0.0.1 port 953
              allow { 127.0.0.1; } keys { "rndckey"; };
 };
```

(4) 重启 named 进程后，就可以使用 rndc 工具对 named 进行控制了。例如，下面的命令可以使 named 重新装载配置文件和区文件：

```
# rndc reload
server reload successful
```

此外，所有 rndc 支持的命令及帮助信息可以通过不带参数的 rndc 命令显示：

```
# rndc
Usage: rndc [-c config] [-s server] [-p port]
       [-k key-file ] [-y key] [-V] command

command is one of the following:

  reload        Reload configuration file and zones.
  reload zone [class [view]]
                Reload a single zone.
  refresh zone [class [view]]
                Schedule immediate maintenance for a zone.
  retransfer zone [class [view]]
                Retransfer a single zone without checking serial number.
  freeze zone [class [view]]
                Suspend updates to a dynamic zone.
  thaw zone [class [view]]
                Enable updates to a frozen dynamic zone and reload it.
  reconfig      Reload configuration file and new zones only.
  stats         Write server statistics to the statistics file.
  querylog      Toggle query logging.
  dumpdb [-all|-cache|-zones] [view ...]
                Dump cache(s) to the dump file (named_dump.db).
  stop          Save pending updates to master files and stop the server.
  stop -p       Save pending updates to master files and stop the server
                reporting process id.
  halt          Stop the server without saving pending updates.
  halt -p       Stop the server without saving pending updates reporting
                process id.
  trace         Increment debugging level by one.
  trace level   Change the debugging level.
  notrace       Set debugging level to 0.
  flush         Flushes all of the server's caches.
  flush [view]  Flushes the server's cache for a view.
  flushname name [view]
                Flush the given name from the server's cache(s)
  status        Display status of the server.
  recursing     Dump the queries that are currently recursing
(named.recursing)
  *restart      Restart the server.

* == not yet implemented
Version: 9.3.6-P1-Red Hat-9.3.6-4.P1.el5_4.2
```

根据上面结果可知，rndc 提供了非常丰富的命令，可以让管理员在不重启 named 进程的情况下，完成大部分的 DNS 服务器管理工作。

6.4 bind-chroot 简介

早期 Linux 服务都是以 root 权限启动和运行的，随着技术的发展，各种服务变得越来越复杂，导致问题和漏洞越来越多。黑客利用服务的漏洞入侵系统，能获得 root 级别的权限，从而控制整个系统。为了减缓这种攻击所带来的负面影响，现在服务器软件通常设计

为以 root 权限启动，然后服务器进程自行放弃 root，再以某个低权限的系统账号来运行进程。这种方式的好处在于该服务被攻击者利用漏洞入侵时，由于进程权限很低，攻击者得到的访问权限又是基于这个较低权限的，因此对系统造成的危害比以前减轻了许多。

chroot 是 Change Root 的缩写，它可以改变程序运行时所参考的“/”根目录位置，即将某个特定的子目录作为程序的虚拟“/”根目录。chroot 对程序运行时可以使用的系统资源、用户权限和所在目录进行严格控制，程序只在这个虚拟的根目录具有权限，一旦跳出该目录就无任何权限了，所以有些书籍也将 chroot 称为“jail 监狱”。举个简单的例子，架设过 FTP 服务器的读者都知道，用户登录到 FTP 服务器时，看到的根目录并不是服务器上真正的根目录，而是它的主目录。用户不能访问除主目录外的任何资源，即将用户“jail 监禁”在自己的主目录中，用户的任何操作仅对自己的主目录有效，不会影响系统和其他用户的文件，chroot 的作用也是类似的。

使用了 chroot 后，就算黑客入侵了某个服务，由于具有的权限相当有限，因此最多也只能破坏该服务的虚拟根目录，不会威胁到整个服务器的安全。对 DNS 服务而言，有经验的网络管理员都会对其使用 chroot 技术来增强 BIND 的安全。在安装 bind-chroot 后，BIND 的工作目录发生变化，更改到/var/named/chroot/目录下，即使服务器 DNS 系统被非法入侵，其影响范围也仅限于/var/named/chroot/目录，不会对整个系统造成影响。

安装 chroot 前，BIND 的主配置文件位于/etc 目录下，域名数据库文件位于/var/named 文件夹下。安装 chroot 后，虚拟根目录变为/var/named/chroot，同时，这个虚拟根目录下还自动创建 etc 和 var 目录，分别对应实际根目录下的同名目录。

另外，安装 chroot 时，还会自动把实际根目录下的对应目录中的配置文件都复制到虚拟根目录下对应的目录中。例如，/etc/named.conf 会复制到/var/named/chroot/etc 文件夹中。因此下文提到所有的 DNS 服务器配置文件、区域数据文件和配置文件内的语句，都是相对这个虚拟根目录而言的。

当 chroot 包安装完后，会在/usr/sbin 目录下创建 bind-chroot-admin 文件，这是 chroot 的命令文件，利用 bind-chroot-admin 命令可以禁用或启用 chroot 功能，也可以使虚拟根目录下的 named 配置文件与实际根目录下的 named 配置文件进行同步。其命令格式如下所示：

```
# bind-chroot-admin
Usage:
  -e | --enable:   enable the bind-chroot environment
  -d | --disable:  disable the bind-chroot environment
  -s | --sync:     sync files between the bind chroot and / environments,
                 so they are correct for the current state of the bind-
chroot
                 (enabled / disabled)
  $BIND_CHROOT_PREFIX, default /var/named/chroot, is the location of the
chroot.
  $BIND_DIR, default /var/named, is the default un-chrooted bind
directory.
```

由上述结果可知，在 bind-chroot-admin 命令后加-e 选项可以启用 chroot 功能，加-d 选项禁用 chroot 功能，加-s 选项同步配置文件。在学习与工作中配置 BIND，最好要启用 chroot 功能，可以使服务器的安全性能得到提高。

注意： 为避免产生混淆，下文中关于 BIND 的配置文件位置均指在安装 bind-chroot 后的文件位置。

6.5 BIND 服务的配置文件

BIND 服务的配置文件比较多，用户可以直接修改配置文件来更改相应的设置。配置文件更改完成后，需要重新启动或重新加载配置文件才能生效。下面介绍几个常用的配置文件。

- BIND 主配置文件：/var/named/chroot/etc/named/named.conf 文件为 BIND 服务的主配置文件，用户可以在此配置文件中设置域名转发、访问控制、视图等。
- BIND 区域配置文件：/var/named/chroot/etc/named.zones 文件是 BIND 服务的另一个主配置文件，用户可以在此配置文件中进行正向区域、反向区域设置。
- BIND 根域配置文件：/var/named/chroot/var/named/named.ca 文件是 BIND 根域配置文件，主要记录了全球根域服务器的 IP 地址和域名。用户可以定期对此文件进行更新。
- BIND 正向域配置文件：/var/named/chroot/var/named/localhost.zone 文件为 BIND 正向区域配置文件模版，主要记录 localhost 域的信息，如主机记录、起始授权记录、域名服务器记录等，用于将域名解析成 IP 地址。同时用户可以以此文件为模版，在创建正向区域数据库文件时，可以直接复制此文件为模版。
- BIND 反向区域配置文件：/var/named/chroot/var/named/named.local 文件为 BIND 反向区域配置文件模版，主要记录 127.0.0 域的信息，如指针记录等，用于将 IP 地址解析成域名。
- BIND 执行脚本：/etc/init.d/named 文件为 BIND 服务器启动、停止的脚本。用户可以直接执行此脚本运行 BIND 服务。如输入/etc/init.d/named start 即可启动 BIND 服务。
- BIND 守护进程：/usr/sbin/named 文件为 BIND 服务的守护进程，是 BIND 服务的执行文件。
- BIND 日志文件：/var/log/messages 文件为 BIND 服务日志文件，同时也是其他服务的日志文件，记录了服务的启动、停止、重启等相关日志。
- BIND 客户端文件：/etc/resolv.conf 是 DNS 客户端文件，记录 DNS 服务器 IP 地址及域名等。

注意： BIND 的主配置文件可以是 named.conf，也可以是 named.caching-nameserver.conf。事实上，named.caching-nameserver.conf 是 named.conf 的高速缓存文件，其优先级高于 named.conf 配置文件。若/var/named/chroot/etc/文件夹不存在 named.conf，用户可以直接将 named.caching-nameserver.conf 改名为 named.conf，或者直接使用 named.caching-nameserver.conf 作为配置文件即可。若两个文件同时存在，则 BIND 会将 named.caching-nameserver.conf 作为其主配置文件。

6.5.1　主要配置文件 named.conf

/var/named/chroot/etc/named.conf 是 BIND 服务的主配置文件。用户可以在此配置文件中设置域名转发、视图、访问控制等。默认内容如下所示：

```
[root@bogon etc]# more named.conf
//
// named.caching-nameserver.conf
//
// Provided by Red Hat caching-nameserver package to configure the
// ISC BIND named(8) DNS server as a caching only nameserver
// (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration
files.
//
// DO NOT EDIT THIS FILE - use system-config-bind or an editor
// to create named.conf - edits to this file will be lost on
// caching-nameserver package upgrade.
//
options {
//设置 BIND 服务的相关选项，如域名转发、BIND 服务器配置文件目录、监听端口等
        listen-on port 53 { 127.0.0.1; };
        listen-on-v6 port 53 { ::1; };
        //设置监听端口为 53
        directory       "/var/named";
        //设置区域数据库文件的存储位置
        dump-file       "/var/named/data/cache_dump.db";
        //设置缓存 dump 文件的存储位置
        statistics-file "/var/named/data/named_stats.txt";
        //设置 BIND 服务器状态文件位置
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        设置 BIND 服务器内存状态文件位置

        // Those options should be used carefully because they disable port
        // randomization
        // query-source    port 53;
        // query-source-v6 port 53;

        allow-query     { localhost; };
        //设置允许 DNS 查询的地址或地址范围，默认允许本机查询
        allow-query-cache { localhost; };
        //设置允许查询缓存的地址或地址范围。
};
logging {
//配置 BIND 日志
        channel default_debug {
        //配置日志通道
                file "data/named.run";
                severity dynamic;
        };
};
view localhost_resolver {
//配置 BIND 视图
        match-clients      { localhost; };
        //客户机地址
```

```
        match-destinations { localhost; };
        //客户端主机
        recursion yes;
        //是否允许递归查询
        include "/etc/named.rfc1912.zones";
        //包含区域文件的位置。
};
```

named.conf 配置文件由语句与注释组成，每一条主配置语句均有自己的选项参数。这些选项参数以子语句的形式组成，并包含在花括号内，作为主语句的组成部分。每一条语句，包括主语句和子语句，都必须以分号结尾。注释符号可以使用类似于 C 语言中的块注释“/*”和“*/”符号对，以及行注释符“//”或“#”。

BIND 支持的主配置语句及功能如表 6-1 所示。

表 6-1 BIND 的主配置语句及功能

主配置语句名称	功 能
acl	定义一个访问控制列表，用于以后对列表中的 IP 进行访问控制
controls	定义有关本地域名服务器操作的控制通道，这些通道被 rndc 用来发送控制命令
include	把另一个文件中的内容包含进来做为主配置文件的内容
key	定义一个密匙信息，用于通过 TSIG 进行授权和认证的配置中
logging	设置日志服务器，以及日志信息的发送位置
options	设置 DNS 服务器的全局配置选项
server	定义了与远程服务器交互的规则
trusted-keys	定义信任的 DNSSED 密匙
view	定义一个视图
zone	定义一个区域

下面分别就各个配置语句作简单介绍：

1) acl 语句

acl 主配置语句用于定义一个访问控制列表，列表中包含了一些用 IP 表示的主机，这个访问列表可以在其他语句中使用，表示其所定义的主机。其格式如下：

```
acl acl-name {      address_match_list   };
```

address_match_list 表示 IP 地址或 IP 地址集。其中，none、any、localhost 和 localnets 这 4 个内定的关键字有特别含义，分别表示没有主机、任何主机、本地网络接口 IP 和本地子网 IP。一个具体的例子如下所示：

```
acl "someips" {            //定义一个名为 someips 的 ACL
    10.0.0.1; 192.168.23.1; 192.168.23.15;     //包含 3 个单个 IP
};
acl "complex" {            //定义一个名为 complex 的 ACL
    "someips";                 //可以包含其他 ACL
    10.0.15.0/24;              //包含 10.0.15.0 子网中的所有 IP
    !10.0.16.1/24;             //排除 10.0.16.1 子网的 IP
    {10.0.17.1;10.0.18.2;};    //包含了一个 IP 组
```

```
        localhost;              //本地网络接口 IP(含实际接口 IP 和 127.0.0.1)
    };
```

2) controls 语句

controls 主语句定义有关本地域名服务器操作的控制通道，这些通道被 rndc 用来发送控制命令。在上节的例子 named.conf 配置文件中有以下语句，现解释如下：

```
    controls {
        inet 127.0.0.1 port 953    //在 127.0.0.1 接口的 953 号端口进行监听
        allow { 127.0.0.1; }    //只接受 127.0.0.1 的连接，即只有在本机使用 rndc 才能
    对 named 进行控制
        keys { "rndckey"; };     //使用名为 rndckey 的密钥才能访问
    };
```

3) include 语句

include 主语句表示把另一个文件的内容包含进来，作为 named.conf 文件的配置内容，其效果与把那个文件的内容直接输入 named.conf 时一样。采用这种方式一是为了简化 named.conf 文件的管理；二是为了安全，因为可以把一些密钥放在其他文件不让无关的人查看。

4) key 语句

key 主语句定义一个密匙，用于 TSIG 授权和认证。它主要在与其他 DNS 服务器或 rndc 工具通信时使用，可以通过运行 rndc-confgen 命令产生。

5) logging 语句

logging 是有关日志配置的主语句，可以有众多的子语句，指明了日志记录的位置、日志的内容、日志文件的大小和日志的级别等内容。下面是一个典型的日志语句内容。

```
    logging{
    channel simple_log {  //定义一个名为 simple_log 的日志通道。可以定义多个通道，每
    个通道代表一种日志
    file "/var/log/named/bind.log" versions 3                 //该日志记录在
    /var/log/named/bind.log 文件中，版本号为 3
    size 5m;      //文件的大小是 5MB，超过 5MB 时，会以 bind.log.1 的名字备份起来
    severity warning;  //高于或等于 warning 级别的日志才被记录
    print-time yes;     //日志记录包含时间域
    print-severity yes; //日志记录包含日志级别域
    print-category yes; //日志记录包含日志分类域    };
    category default{     //所有的分类都记录到 simple_log 日志通道中
    simple_log;
    };
    };
```

6) options 语句

options 语句设定可以被整个 BIND 使用的全局选项。这个语句在每个配置文件中只有一处，如果出现多个 options 语句，则第一个 options 的配置有效，并且会产生一个警告信息。如果没有 options 语句，每个子语句使用默认值。options 选项的子语句很多，下面简单介绍一下主配置文件中的子语句：

- directory：指定服务器的工作目录。配置文件其他语句中所使用的相对路径，指的都是在这个子语句指定的目录下。大多数的输出文件默认时也生成在这个目录

下。如果没有设定，工作目录默认设置为服务器启动时的目录。指定目录时，应该以绝对路径表示。

- pid-file：设定进程 PID 文件的路径名，如果没有指定，默认为/var/run/named.pid。因此，此时要注意运行进程的用户 named 对该目录要有写入的权限，否则，named 将不能正常启动。pid-file 是给那些需要向运行着的服务器发送信号的程序使用的。
- forwarders：设定转发使用的 IP 地址。该子语句只有在 forward 设置成允许转发后才生效，默认的列表是空的，表示不转发。转发也可以设置在每个域中，这样全局选项中的转发设置就不会起作用了。用户可以将不同的域转发到不同的其他 DNS 服务器上，或者对不同的域实现 forward only 或 first 的不同方式，也可以选择根本就不转发。
- allow-query：主语句用于设定 DNS 服务器为哪些客户机提供 DNS 查询服务，可以在后面的花括号内放置命名的 ACL 或 address_match_list，any 表示任何主机都可以访问。allow-query 也能在 zone 语句中设定，这样全局 options 中的 allow-query 选项在 zone 中就不起作用了。默认时是允许所有主机进行查询。

7) server 语句

server 主语句定义了与远程服务器交互的规则，例如，决定本地 DNS 服务器是作为主域名服务器还是辅域名服务器，以及与其他 DNS 服务器通信时采用的密钥等。语句可以出现在配置文件的顶层，也可以出现在视图语句的内部。如果一个视图语句包括了自己的 server 语句，则只有那些视图语句内的 server 语句才起作用，顶层的 server 语句将被忽略。如果一个视图语句内不包括 server 语句，则顶层 server 语句将被当做默认值。

8) trusted-keys 语句

trusted-keys 语句定义 DNSSEC 安全根的 trusted-keys。DNSSEC 指由 RFC2535 定义的 DNS sercurity。当一个非授权域的公钥是已知的，但不能安全地从 DNS 服务器获取时，需要加入一个 trusted-keys。这种情况一般出现在 singed 域是一个非 signed 域的子域的时候，此时加了 trusted key 后被认为是安全的。trusted-keys 语句能包含多重输入口，由键的域名、标志、协议算法和 64 位键数据组成。

9) view 语句

view 语句定义了视图功能。视图是 BIND 9 提供的强大的新功能，允许 DNS 服务器根据客户端的不同有区别地回答 DNS 查询，每个视图定义了一个被特定客户端子集见到的 DNS 名称空间。这个功能在一台主机上运行多个形式上独立的 DNS 服务器时特别有用。

6.5.2　主要配置文件 named.rfc.zones

/var/named/chroot/etc/named.rfc.zones 也是 BIND 服务的主配置文件，用户可以在此配置文件中设置域名正向解析、反向解析，默认内容如下(以//为注释语句)：

```
# more named.rfc1912.zones
// named.rfc1912.zones:
//
// Provided by Red Hat caching-nameserver package
//
// ISC BIND named zone configuration for zones recommended by
```

```
// RFC 1912 section 4.1 : localhost TLDs and address zones
//
// See /usr/share/doc/bind*/sample/ for example named configuration
files.
//
zone "." IN {              //设置根区域
        type hint;         //区域的类型为根域
        file "named.ca";  //根域对应的区域数据库文件
};

zone "localdomain" IN {  //定义 localdomain 区域，域名为 localdoman
        type master;       //表示区域类型为主域名。
        file "localdomain.zone";  //区域数据库文件的文件名，记录区域的相关记录。
        allow-update { none; };  //允许客户端进行更新的地址范围。
};


zone "localhost" IN {
//设置 localhost 正向区域，如需创建域名，则可将域名替换 localhost
        type master;
        file "localhost.zone";  //此域的区域文件位置。
        allow-update { none; };
};

zone "0.0.127.in-addr.arpa" IN {
//设置 127.0.0 反向区域，其中 IP 地址需要反写。
        type master;
        file "named.local";
        allow-update { none; };
};

zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa"
IN  //设置一个 ipv6 的反向区域
 {
        type master;
        file "named.ip6.local";
        allow-update { none; };
};

zone "255.in-addr.arpa" IN {  //设置一个广播域
        type master;
        file "named.broadcast";
        allow-update { none; };
};

zone "0.in-addr.arpa" IN {     //设置一个零区域
        type master;
        file "named.zero";
        allow-update { none; };
};
```

zone 语句定义了 DNS 服务器所管理的区，它定义了哪些域名是授权给该 DNS 服务器回答的。type 子语句可指定 5 种类型的区，具体名称和功能如下所示：

- Master(主域)：主域用来保存某个区域的数据信息。
- Slave(辅助区域)：也叫次级域，数据来自主域，起备份作用。
- Stub：Stub 区与辅域相似，但它只复制主域的 NS 记录，而不是整个区数据。它

不是标准 DNS 的功能，只是 BIND 提供的功能。

- Forward(转发)：转发域中一般配置了 forward 和 forwarders 子句，用于把对该域的查询请求转由其他 DNS 服务器处理。
- Hint：Hint 域定义了一套最新的根 DNS 服务器地址，如果没有定义，DNS 服务器会使用内建的根 DNS 服务器地址。

在每一个 zone 语句中，都用 file 子语句定义一个区文件，这个文件里存放了域名与 IP 地址的对应关系。

6.5.3 正向区域数据库文件

该文件用于区域域名解析查询，内容模板如下：

```
# more /var/named/chroot/var/named/localhost.zone
$TTL    86400
//TTL 为 Time to live 的缩写，即该区域数据库文件的生存周期，单位为秒。通常所有的区域数据库文件设置默认相同，86400 秒即为 1 天。
@              IN SOA  @       root (
//设置区域的起始授权记录。其中第一个@表示域名，第二个@表示起始授权机构即 DNS 的 IP 地址。root 表示该域管理员邮箱。假如管理员邮箱地址为 nic@edu.cn，则需要在此处写成 nic.edu.cn。因为@符号在本文件中已经有了其他的定义。
                    42              ; serial (d. adams)
                    //表示序列号，如果区域数据库文件的内容发生了更改，则需将此数值加 1，在辅助域名服务器更新记录时使用。若本身的序列号比主域名服务器的序号小，则更新自身的记录。
                     3H              ; refresh
                            //域名服务器刷新记录的时间为 3 小时，辅助域名服务器每隔 3 小时向主域名服务器发出一次更新请求。
                                  15M             ; retry
                            //当辅助域名服务器 3 小时后无法与主域名服务器通信，将每隔 15 分钟再向主域名服务器发送更新请求。
                                  1W              ; expiry
                            //过期时间为 1 周。若辅助域名服务器 1 周时间无法与主域名服务器通信，则对应的记录将失效。
                                  1D )            ; minimum
                            //TTL 的最小值为 1 天。

              IN NS          @
                //NS 记录，表示域名服务器记录对应的主机域名。
              IN A           127.0.0.1
                //A 记录，记录主机与域名的映射关系。默认主机名位 localhsot 对应的 IP 地址为 127.0.0.1。
              IN AAAA        ::1
                //IPv6 的主机 AAAA 记录。
```

在区域数据文件中，使用";"作为行注释符，除第一条语句以外，区域数据文件中的每一条语句称为一条记录。以上配置中各条语句的含义如下所示。

1) 设置其他 DNS 服务器缓存本机数据的默认时间

$TTL 指令要求放在文件的第 1 行，定义了其他 DNS 服务器缓存本机数据的默认时间，默认单位是秒，也可以用 h(小时)、d(天)和 w(星期)为单位。DNS 服务器在应答中提供 TTL 值，目的是允许其他的服务器在 TTL 间隔内缓存数据。如果本地的 DNS 服务器数据改变不大，可以考虑几天默认 TTL，最长可以设为一周。但是不推荐设置 TTL 为

0，此时将导致大量的 DNS 数据传输。

2) 设置起始授权机构

SOA 是 Start of Authority(起始授权机构)的缩写，它指出这个域名服务器是作为该区数据的权威的来源。在指令“testedu.cn. IN SOA ns.testedu.cn. nic@testedu.cn.”中，指定了负责解析 testedu.cn.域的授权主机名是“ns.testedu.cn.”，授权主机名称将在区域文件中解析为 IP 地址。IN 表示属于 Internet 类，是固定不变的，“n@testedu.cn.”表示负责该区域的管理员的 E-mail 地址。每一个区文件都需要一个 SOA 记录，而且只能有一个。SOA 资源记录还要指定一些附加参数，放在 SOA 资源记录后面的括号内，其名称和功能见例子中的注释。

3) 设置名称服务器 NS 资源记录

例如，“testedu.cn. IN NS ns.testedu.cn.”是一条 NS(Name Server)资源记录，定义了域“testedu.cn.”由 DNS 服务器“ns.testedu.cn.”负责解析，NS 资源记录定义的服务器称为区域权威名称服务器。权威名称服务器负责维护和管理所管辖区域中的数据，被其他服务器或客户端当作权威的来源，并且能肯定应答区域内所含名称的查询。这里的配置要求和 SOA 记录配置一致。

4) 设置邮件服务器 MX 资源记录

例如，“testedu.cn. IN MX 10 mail”是一条 MX(Mail eXchanger)资源记录，表示发往 testedu.cn 域的电子邮件由 mail.testedu.cn 邮件服务器负责处理。例如，当一个邮件要发送地址到 test@testedu.cn 时，发送方的邮件服务器通过 DNS 服务器查询 testedu.cn 这个域名的 MX 资源记录，查到后，会把邮件发送到指定的邮件服务器，如 mail.testedu.cn。至于该域名对应的 IP 地址，需要通过随后的 A 资源记录设定。

值得注意的是，在设置邮件服务器 MX 资源记录的时候可以设置多个 MX 资源记录，指明多个邮件服务器，优先级别由 MX 后的数字决定，数字越小，邮件服务器的优先权越高。邮件传送时首先选用优先级高的邮件服务器，当邮件传送给优先级高的邮件服务器失败时，将邮件传送给优先级低的邮件服务器。

5) 设置主机地址 A 资源记录

主机地址 A(Address)资源记录是最常用的记录，它定义了 DNS 域名对应 IP 地址的信息。在主机域名部分可以直接写相对名称，也可以写完全规范域名 FQDN，两者的功能是一样的。

6) 设置别名 CNAME 资源记录

别名 CNAME(Canonical Name)资源记录也被称为规范名字资源记录。CNAME 资源记录允许将多个名称映射到同一台计算机上，使得某些任务更容易执行。例如，对于同时提供 Web、OA 服务的计算机(IP 地址为 10.10.1.3)，为了便于用户访问服务，可以先为其建立一条主机地址 A 资源记录“www IN A 10.10.1.3”，将 www.testedu.cn 映射到 10.10.1.3 地址，然后再为该计算机设置 oa 别名，即建立 CNAME 资源记录“oa IN CNAME www”。这样，当访问 www.testedu.cn 和 oa.testedu.cn 时，实际都是访问 IP 地址为 10.10.1.3 的计算机。

6.5.4　反向区域数据库文件

用于反向区域域名解析。具体模板内容如下：

```
# more /var/named/chroot/var/named/named.local
$TTL    86400
$TTL    86400
//TTL 为 Time to live 的缩写，即该区域数据库文件的生存周期，单位为秒。通常所有的区域
数据库文件设置默认相同，86400 秒即为 1 天。
@              IN SOA  @       root (
//设置区域的起始授权记录。其中第一个@表示域名，第二个@表示起始授权机构即 DNS 的 IP 地
址，root 表示该域管理员邮箱。假如管理员邮箱地址为 nic@edu.cn，则需要在此处写成
nic.edu.cn。因为@符号在本文件中已经有了其他的定义。
                                   42              ; serial (d. adams)
                                     //表示序列号，如果区域数据库文件的内容发生了
更改，则需将此数值加 1，在辅助域名服务器更新记录时使用。若本身的序列号比主域名服务器的
序号小，则更新自身的记录。
                                   3H              ; refresh
                                     //域名服务器刷新记录的时间为 3 小时，辅助域名
服务器每隔 3 小时向主域名服务器发出一次更新请求。
                                   15M             ; retry
                                     //当辅助域名服务器 3 小时后无法与主域名服务器
通信，将每隔 15 分钟再向主域名服务器发送更新请求。
                                   1W              ; expiry
                                     //过期时间为 1 周。若辅助域名服务器 1 周时间
无法与主域名服务器通信，则对应的记录将失效。
                                   1D )            ; minimum
                                     //TTL 的最小值为 1 天。

              IN NS          @
                //NS 记录，表示域名服务器记录对应的主机域名
1       IN      PTR     localhost.
//PTR 指针记录，用于将 IP 地址解析成域名。
```

反向域名解析是通过 in-addr.arpa 域和 PTR 记录实现的。in-addr.arpa 域从左至右阅读，这与 IP 地址的通常顺序相反。于是，一台 IP 地址为 10.16.2.3 的机器将会有对应的 in-addr.arpa 名称：3.2.16.10.in-addr.arpa。这个名称应该具有一个 PTR 资源记录，它的数据字段是主机名称。下面看一下以上配置的具体解释。

1) 设置 SOA 和 NS 资源记录

反向解析区域文件必须包括 SOA 和 NS 资源记录，使用固定格式的反向解析区域 in-addr.arpa 作为域名，结构和格式与区域数据文件类似，这里不再重复。

2) 设置指针 PTR 资源记录

指针 PTR 资源记录只能在反向解析区域文件中出现。PTR 资源记录和 A 资源记录正好相反，它是将 IP 地址解析成 DNS 域名的资源记录。与区域文件的其他资源记录类似，它也可以使用相对名称和完全规范域名 FQDN。例如，“6.1.16.10.in-addr.arpa. IN PTR mail.testedu.cn.”表示 IP 地址 10.16.1.6 对应的域名为 mail.testedu.cn。

6.5.5 根域数据库文件

该文件记录了全球根域服务器的 IP 地址，包括 13 台 IPv4 的根域服务器地址和 6 台 IPv6 的根域服务器 IP 地址。当 BIND 接到客户端的查询请求时，如果本地不能解释，也不能在 Cache 中找到相应的数据，就会通过根服务器进行逐级查询。具体内容如下：

```
# more /var/named/chroot/var/named/named.ca

; <<>> DiG 9.5.0b2 <<>> +bufsize=1200 +norec NS . @a.root-servers.net
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7033
;; flags: qr aa; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 20

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;.                          IN     NS

;; ANSWER SECTION:
.                    518400  IN     NS     D.ROOT-SERVERS.NET.
.                    518400  IN     NS     E.ROOT-SERVERS.NET.
.                    518400  IN     NS     F.ROOT-SERVERS.NET.
.                    518400  IN     NS     G.ROOT-SERVERS.NET.
.                    518400  IN     NS     H.ROOT-SERVERS.NET.
.                    518400  IN     NS     I.ROOT-SERVERS.NET.
.                    518400  IN     NS     J.ROOT-SERVERS.NET.
.                    518400  IN     NS     K.ROOT-SERVERS.NET.
.                    518400  IN     NS     L.ROOT-SERVERS.NET.
.                    518400  IN     NS     M.ROOT-SERVERS.NET.
.                    518400  IN     NS     A.ROOT-SERVERS.NET.
.                    518400  IN     NS     B.ROOT-SERVERS.NET.
.                    518400  IN     NS     C.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
A.ROOT-SERVERS.NET.     3600000 IN     A      198.41.0.4
A.ROOT-SERVERS.NET.     3600000 IN     AAAA    2001:503:ba3e::2:30
B.ROOT-SERVERS.NET.     3600000 IN     A      192.228.79.201
C.ROOT-SERVERS.NET.     3600000 IN     A      192.33.4.12
D.ROOT-SERVERS.NET.     3600000 IN     A      128.8.10.90
E.ROOT-SERVERS.NET.     3600000 IN     A      192.203.230.10
F.ROOT-SERVERS.NET.     3600000 IN     A      192.5.5.241
F.ROOT-SERVERS.NET.     3600000 IN     AAAA    2001:500:2f::f
G.ROOT-SERVERS.NET.     3600000 IN     A      192.112.36.4
H.ROOT-SERVERS.NET.     3600000 IN     A      128.63.2.53
H.ROOT-SERVERS.NET.     3600000 IN     AAAA    2001:500:1::803f:235
I.ROOT-SERVERS.NET.     3600000 IN     A      192.36.148.17
J.ROOT-SERVERS.NET.     3600000 IN     A      192.58.128.30
J.ROOT-SERVERS.NET.     3600000 IN     AAAA    2001:503:c27::2:30
K.ROOT-SERVERS.NET.     3600000 IN     A      193.0.14.129
K.ROOT-SERVERS.NET.     3600000 IN     AAAA    2001:7fd::1
L.ROOT-SERVERS.NET.     3600000 IN     A      199.7.83.42
M.ROOT-SERVERS.NET.     3600000 IN     A      202.12.27.33
M.ROOT-SERVERS.NET.     3600000 IN     AAAA    2001:dc3::35

;; Query time: 110 msec
;; SERVER: 198.41.0.4#53(198.41.0.4)
;; WHEN: Tue Feb 26 15:05:57 2008
;; MSG SIZE  rcvd: 615
```

一般来说，根域文件无需进行修改。有时 Internet 根服务器的地址会发生变化，因此 named.ca 也应该随之更新。最新的根服务器列表可以从 ftp://ftp.rs.internic.net/domain/named.root 下载，它包含了国际互联网络信息中心(InterNIC)提供的最新数据。

6.5.6 日志文件

BIND 服务的日志文件为/var/log/messages，主要记录有效的区域、区域数据库文件的序号、BIND 服务器的运行状态等。如果用户需要记录 BIND 服务的其他日志，可在 BIND 服务器上通过 logging 选项进行配置。通过查看该文件可得到如下内容：

```
Mar 10 08:00:49 bogon named[10253]: starting BIND 9.3.6-P1-RedHat-9.3.6-
4.P1.el5_4.2 -u named -c /etc/named.cach
ing-nameserver.conf -t /var/named/chroot
Mar 10 08:00:49 bogon named[10253]: adjusted limit on open files from
1024 to 1048576
Mar 10 08:00:49 bogon named[10253]: found 2 CPUs, using 2 worker threads
Mar 10 08:00:49 bogon named[10253]: using up to 4096 sockets
Mar 10 08:00:49 bogon named[10253]: loading configuration from
'/etc/named.caching-nameserver.conf'
Mar 10 08:00:49 bogon named[10253]: using default UDP/IPv4 port range:
[1024, 65535]
Mar 10 08:00:49 bogon named[10253]: using default UDP/IPv6 port range:
[1024, 65535]
Mar 10 08:00:49 bogon named[10253]: listening on IPv4 interface eth0,
192.168.0.104#53
Mar 10 08:00:49 bogon named[10253]: command channel listening on
127.0.0.1#953
Mar 10 08:00:49 bogon named[10253]: command channel listening on ::1#953
Mar 10 08:00:49 bogon named[10253]: the working directory is not
writable
Mar 10 08:00:49 bogon named[10253]: zone 0.in-
addr.arpa/IN/localhost_resolver: loaded serial 42
Mar 10 08:00:49 bogon named[10253]: zone 0.0.127.in-
addr.arpa/IN/localhost_resolver: loaded serial 1997022700
…(省略)
```

6.6 BIND 服务器常用调试工具

为了方便用户对 BIND 进行配置，减少手工修改配置文件而产生的错误，BIND 中也集成了一些服务器配置工具。下面，让我们对几个工具做以下介绍。

6.6.1 配置文件语句检测工具

named-checkconf 工具用来检测 BIND 主配置文档 named.conf 是否存在错误。命令运行可以参照以下模式：

```
# named-checkconf  /var/named/chroot/etc/named.conf
```

如主配置文件 named.conf 没有错误，则不返回数据。如存在错误，则出现错误提示如下所示：

```
# named-checkconf  /var/named/chroot/etc/named.conf
/var/named/chroot/etc/named.conf:18: missing ';' before 'directory'
```

以上结果显示，在/var/named/chroot/etc/named.conf 文件的第 18 行在'directory'字符之

前缺少“;”。

6.6.2　区域数据库文件语句检测工具

named-checkzone 工具用来检测 BIND 区域数据库文件是否存在语法错误。命令语法格式为：

```
named-checkzone <区域> <区域数据库文件>
```

如需要检查 testedu.cn 正向区域数据库是否存在错误，可用以下命令实现：

```
# named-checkzone testedu.cn /var/named/chroot/var/named/db.testedu.cn
```

运行结果如下所示：

```
# named-checkzone testedu.cn /var/named/chroot/var/named/db.testedu.cn
zone testedu.cn/IN: loaded serial 42
OK
```

说明数据库文件没有语法错误。

6.7　DNS 客户端的配置

6.7.1　Linux 中 DNS 客户端的配置

在 Linux 中为客户端指定 DNS 服务器地址，可直接编辑配置文件/etc/resolv.conf。/etc/resolv.conf 文件是 Linux 客户端定义 DNS 服务器的配置文件。当客户端需要解析某个域名时，会将此解析的查询请求直接发送到客户端定义的 BIND 服务器的 IP 地址处，要求 BIND 服务器解析。BIND 服务器将解析之后的结果传输给客户端。如果不指定 BIND 服务器的 IP 地址或者 IP 地址错误，则会出现解析不成功的现象。

/etc/resolv.conf 中内容如下：

```
# more /etc/resolv.conf
; generated by /sbin/dhclient-script
search domain.org
nameserver 192.168.1.1
```

其中 search 语句用于设置搜索域。例如 search domain.org 表示设置搜索域为 domain.org 的域名。namedserver 语句用于设置 BIND 服务器的 IP 地址，用于 DNS 客户端解析。可用 nameserver 语句来指定 3 台 DNS 服务器。客户端是按照 DNS 服务器在文件中的顺序进行查询的，如果没有接收到 DNS 服务器的响应，就去尝试向下一台服务器查询，直到试完所有的服务器为止，所以应该将速度最快、最可靠的 DNS 服务器列在最前面，以保证在查询时不会超时。

6.7.2　Windows 中 DNS 客户端的配置

Windows 下配置 DNS 客户端的方法比较简单，在 Windows 环境下设置 DNS 方法大同小异，下面就以配置 Windows 7 的 DNS 客户端为例来说明具体的操作步骤。

(1) 选择“开始”→“控制面板”→“网络和共享中心”，打开图 6-7 所示的“网络和共享中心”窗口。



图 6-7　Windows 7 的“网络和共享中心”窗口

(2) 单击需要配置 DHCP 的网卡所对应的本地连接，打开如图 6-8 所示“本地连接属性”对话框。

(3) 双击“Internet 协议版本 4(TCP/IPv4)”打开 IP 地址配置对话框，如图 6-9 所示。



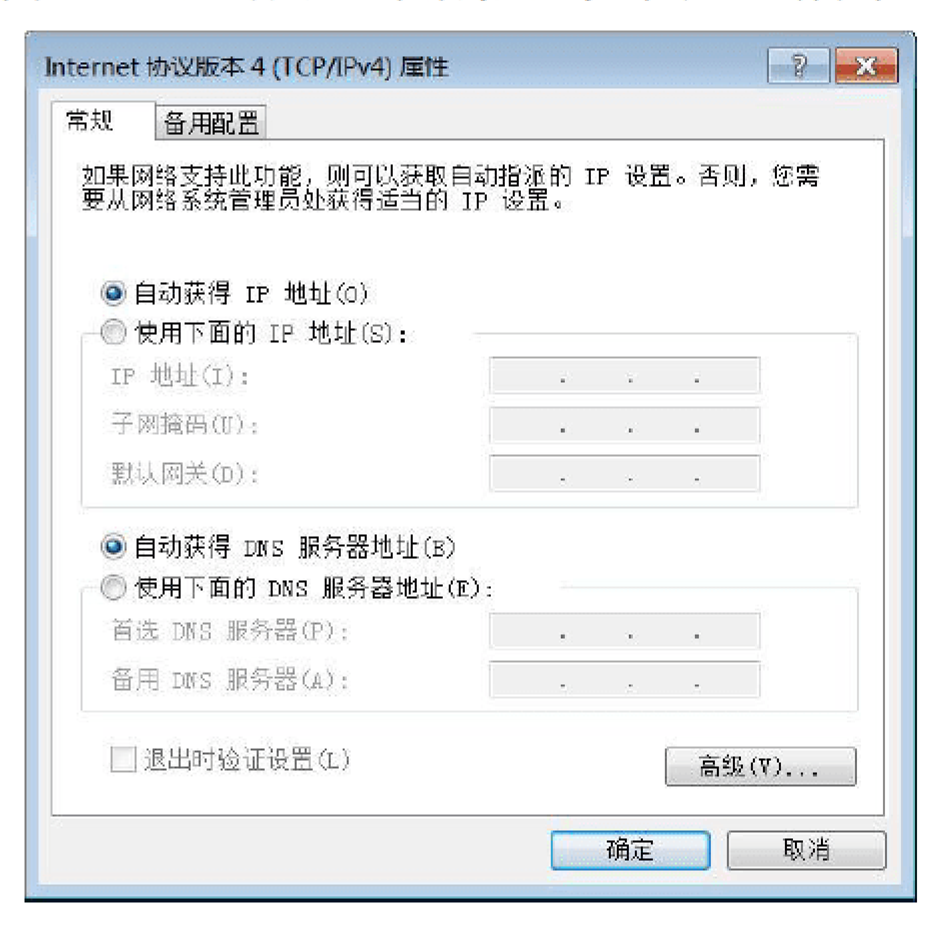


图 6-8　Windows 7 的“本地连接属性”对话框　　图 6-9　“Internet 协议版本 4(TCP/IPv4)属性”对话框

(4) 选中“使用下面的 DNS 服务器地址”单选按钮，在“首选 DNS 服务器”和“备用 DNS 服务器”中输入 DNS 服务器的 IP 地址，然后单击“确定”按钮。

(5) 单击“确定”按钮，即可完成 Windows 7 下的 DNS 客户端的配置。

6.8　BIND 域名服务器的配置步骤

BIND 服务器的配置方法有两种：一种是基于普通模式，另一种是基于 chroot 模式。因 chroot 模式具有更好的安全性，故下面主要针对 chroot 模式介绍 BIND 服务的配置步骤。

(1) 查询是否已经安装 bind-chroot*软件包，如未安装，则安装此软件包。

(2) 修改/var/named/chroot/etc/named.conf 主配置文件，修改其相应的授权。

(3) 修改/var/named/chroot/etc/named.rfc1912.zones 文件，添加其对应的正向、反向区域。

(4) 在/var/named/chroot/var/named 目录下创建区域所对应的数据库文件，并向数据库文件中添加所需的记录。

(5) 启动 BIND 服务。

(6) 修改 BIND 客户端配置文件，指定 BIND 服务器。

(7) 测试域名服务器。

6.9　BIND 主域名服务器配置案例

本节通过测试案例，主要讲解正向域名解析配置、反向域名解析配置、域名负载均衡配置、域名直接解析配置、泛域名解析配置等。

6.9.1　正向域名解析配置

【例 6-1】配置一个正向 DNS 域名服务器，要求域名为 testedu.cn，DNS 服务器地址为 192.168.0.104，DNS 服务器主机名为 ns.testedu.cn。同时要求 DNS 能够为主页服务器 192.168.0.110(www.testedu.cn)、邮件服务器 192.168.0.111(mail.testedu.cn，pop.testedu.cn，smtp.testedu.cn)、FTP 服务器 192.168.0.112(ftp.testedu.cn)和文件服务器 192.168.0.113 (file.testedu.cn)进行解析。

具体操作步骤如下所示。

(1) 进入到配置文件目录，打开 named.conf 配置文件：

```
# cd /var/named/chroot/etc/          //进入到配置文件目录
# vi named.conf                      //用 vi 编辑器打开配置文件
```

修改 named.conf 主配置文件内容：(带注释语句为修改内容)

```
options {
        listen-on port 53 { 192.168.0.104; }; //修改为本地 IP 地址
        //      listen-on-v6 port 53 { ::1; };  //禁用 IPv6 监听
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        allow-query     { any; };              //将 localhost 更改为 any
        allow-query-cache { localhost; };
};
logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};
view localhost_resolver {
```

```
        match-clients      { any; };              //将 localhost 更改为 any
        match-destinations { any; };              //将 localhost 更改为 any
        recursion yes;
        include "/etc/named.rfc1912.zones";
};
```

(2) 修改 named.rfc1912.zones 配置文件，并在文件末尾添加如下内容：

```
zone "testedu.cn" IN {
    type master;
    file "db.testedu.cn";
};
```

(3) 创建正向区域数据库文件，区域数据库文件名为 db.testedu.cn，并对其进行编辑：

```
# cd /var/named/chroot/var/named/
# cp localhost.zone  db.testedu.cn
# vi db.testedu.cn
```

(4) 修改 db.testedu.cn 区域数据库文件，修改内容如下：

```
$TTL    86400
@              IN SOA  @   ns.testedu.cn.    (
                                42              ; serial (d. adams)
                                3H              ; refresh
                                15M             ; retry
                                1W              ; expiry
                                1D )            ; minimum

@       IN      NS      ns.testedu.cn.
ns.testedu.cn.      IN      A       192.168.0.104
www.testedu.cn.     IN      A       192.168.0.110
mail.testedu.cn.    IN      A       192.168.0.111
smtp.testedu.cn.    IN      A       192.168.0.111
pop.testedu.cn.     IN      A       192.168.0.111
ftp.testedu.cn.     IN      A       192.168.0.112
file.testedu.cn.    IN      A       192.168.0.113
@       IN      MX      10      mail.testedu.cn.
```

(5) 更改区域数据库文件所属的组，使其能够正确地被 named 进程读取：

```
# chgrp named db.testedu.cn
```

(6) 启动 DNS 服务，检查 DNS 启动是否正常：

```
# service named start
启动 named:                                                 [确定]

# netstat -an | grep :53
tcp        0      0 192.168.47.129:53         0.0.0.0:*
LISTEN
udp        0      0 192.168.47.129:53         0.0.0.0:*
udp        0      0 0.0.0.0:5353             0.0.0.0:*
udp        0      0 :::5353                 :::*
```

(7) 切换至 DNS 客户端所在机器，修改 DNS 客户端文件：

```
# echo "nameserver 192.168.0.104" > /etc/resolv.conf
```

(8) 测试主机与域名

```
# host  -l testedu.cn
```

显示结果如下。由显示结果可知，已成功将域名解析成对应的 IP 地址。

```
testedu.cn name server ns.testedu.cn.
file.testedu.cn has address 192.168.0.113
ftp.testedu.cn has address 192.168.0.112
mail.testedu.cn has address 192.168.0.111
ns.testedu.cn has address 192.168.0.104
pop.testedu.cn has address 192.168.0.111
smtp.testedu.cn has address 192.168.0.111
www.testedu.cn has address 192.168.0.110
```

(9) 测试邮件交换服务器

```
# host  -t mx testedu.cn
```

显示结果如下所示：

```
testedu.cn mail is handled by 10 mail.testedu.cn.
```

由结果可知，邮件服务器的级别 10，邮件服务器域名为 testedu.cn，邮件服务器主机名为 mail.testedu.cn。

注意： 在区域数据库文件中，所采用的域名应为完全规范域名(FQDN，Fully Qualified Domain Name)即主机名+域名+点号，也就是说每一个域名后面都会有一个点号。

6.9.2　反向域名解析配置

【例 6-2】配置一个反向 DNS 域名服务器，要求直接将输入的 IP 地址解析成域名，域名为 testedu.cn。BIND 服务器地址为 192.168.0.104，BIND 服务器主机名为 ns.testedu.cn。同时要求 BIND 服务能为例 6-1 中的各个域名提供逆向解析服务。

具体操作如下：

(1) 打开/etc/named.rfc1912.zones 配置文件进行编辑：

```
# cd /var/named/chroot/etc/
# vi named.rfc1912.zones
```

(2) 修改 named.rfc1912.zones 文件，文件末尾添加内容如下：

```
zone"0.168.192.in-addr.arpa" IN{
       type master;
       file"rev.testedu.cn";

};
```

(3) 创建反向区域数据库文件，区域数据库文件名为 rev.testedu.cn：

```
# cd /var/named/chroot/var/named
# cp  named.local  rev.testedu.cn
# vi rev.testedu.cn
```

(4) 修改 rev.testedu.cn 文件，修改内容如下：

```
$TTL    86400
@       IN      SOA     ns.testedu.cn.  root.testedu.cn. (
                                        1997022700 ; Serial
                                        28800      ; Refresh
                                        14400      ; Retry
                                        3600000    ; Expire
                                        86400 )    ; Minimum
@     IN      NS      ns.testedu.cn.
104   IN      PTR      ns.testedu.cn.
110   IN      PTR      www.testedu.cn.
111   IN      PTR      mail.testedu.cn.
111   IN      PTR      smtp.testedu.cn.
111   IN      PTR      pop.testedu.cn.
112   IN      PTR      ftp.testedu.cn.
113   IN      PTR      file.testedu.cn.
```

(5) 更改区域数据库文件所属的组，使其能够正确地被 named 进程读取：

```
# chgrp named rev.testedu.cn
```

(6) 重启 BIND 服务，检查 BIND 是否启动成功：

```
# service named restart
停止 named:                                                  [确定]
启动 named:                                                  [确定]
# netstat -an | grep :53
tcp        0      0 192.168.47.129:53         0.0.0.0:*
LISTEN
udp        0      0 192.168.47.129:53         0.0.0.0:*
udp        0      0 0.0.0.0:5353             0.0.0.0:*
udp        0      0 :::5353                  :::*
```

(7) 修改 DNS 客户端配置文件：

```
echo "nameserver 192.168.47.129" > /etc/resolv.conf
```

(8) 解析反向域名测试：

```
# host 192.168.0.104
104.0.168.192.in-addr.arpa domain name pointer ns.testedu.cn.

# host 192.168.0.111
111.0.168.192.in-addr.arpa domain name pointer smtp.testedu.cn.
111.0.168.192.in-addr.arpa domain name pointer pop.testedu.cn.
111.0.168.192.in-addr.arpa domain name pointer mail.testedu.cn.
```

由以上结果可知，通过 IP 地址已经成功地解析出其所对应的域名。在进行反向域名配置时，也需要注意区域数据库文件中的域名应采用完全规范域名。

6.9.3 域名负载均衡配置

目前网络的规模与用户数量急剧膨胀，网络服务器的负担也变得越来越重。一台服务器要同时应付大量用户的并发访问，必然会出现服务器运行效率变低，响应时间过长的结果。通过 DNS 可以简单实现负载均衡的功能。DNS 负载均衡的优点是经济简单易行，它在 DNS 服务器中为同一个域名配置多个 IP 地址(即为一个主机名设置多条 A 资源记录)，

在应答 DNS 查询时，DNS 服务器对每个查询将以 DNS 文件中主机记录的 IP 地址按随机顺序返回不同的解析结果，将客户端的访问引导到不同的计算机上去，使得不同的客户端访问不同的服务器，从而达到负载均衡的目的。

例如，将 www.testedu.cn 域名对应 3 台 WWW 服务器主机，通过配置域名负载均衡，随即进行域名解析。例如，用户 A 访问 www.testedu.cn 域名时，域名服务器可能解析出第 3 台服务器的 IP 地址让其访问；用户 B 访问域名时，域名服务器可能解析第 1 台服务器 IP；用户 C 访问时，域名服务器可能解析出第 2 台服务器 IP，通过这种方式将客户端的集中访问分散到 3 台不同的服务器上，起到了一定的负载均衡作用。

但是域名的负载均衡并不是真正意义上的负载均衡，它无法有效的检测 3 台服务器的负载情况。可能会出现第 2 台服务器经常被解析而第 1 台服务器很少被解析的情况，甚至服务器宕机后不能及时作出改变，出现客户端时而能访问网站，时而又不能访问网站的情况。

下面通过实例的方式讲解域名负载均衡的配置。假设 www.testedu.cn 的 3 台 WWW 服务器主机 IP 地址分别是：192.168.0.110、192.168.0.120、192.168.0.130。可以采用以下方式进行配置，具体配置步骤如下。

编辑区域数据库文件“db.testedu.cn”，文件中加入以下语句：

```
www.testedu.cn.    IN     A      192.168.0.110
www.testedu.cn.    IN     A      192.168.0.120
www.testedu.cn.    IN     A      192.168.0.130
```

重启 BIND 服务：

```
# service named restart
停止 named:                                              [确定]
启动 named:                                              [确定]
```

使用 host 命令进行测试：

```
# host www.testedu.cn
```

返回结果如下所示：

```
www.testedu.cn has address 192.168.0.110
www.testedu.cn has address 192.168.0.120
www.testedu.cn has address 192.168.0.130
```

以上结果表明使用 www.testedu.cn 域名能够解析出 3 个不同的 IP 地址，DNS 负载均衡已经配置成功。

使用 ping 命令进行测试：

```
# ping www.testedu.cn
PING www.testedu.cn (192.168.0.130) 56(84) bytes of data.

--- www.testedu.cn ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2012ms
```

第 1 次测试解析到了 192.168.0.130 的 IP 地址。

```
[root@bogon named]# ping www.testedu.cn
PING www.testedu.cn (192.168.0.120) 56(84) bytes of data.
```

```
--- www.testedu.cn ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 999ms
```

第 2 次测试解析到了 192.168.0.120 的 IP 地址。

```
[root@bogon named]# ping www.testedu.cn
PING www.testedu.cn (192.168.0.110) 56(84) bytes of data.

--- www.testedu.cn ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 999ms
```

第 3 次测试解析到了 192.168.0.110 的 IP 地址。

```
# ping www.testedu.cn
PING www.testedu.cn (192.168.0.130) 56(84) bytes of data.

--- www.testedu.cn ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 1999ms
```

第 4 次测试解析到了 192.168.0.130 的 IP 地址。

```
# ping www.testedu.cn
PING www.testedu.cn (192.168.0.120) 56(84) bytes of data.
```

第 5 次测试解析到了 192.168.0.120 的 IP 地址。

```
--- www.testedu.cn ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 1999ms
```

由以上结果可知，DNS 均衡配置已经生效，配置完成。

6.9.4 域名直接解析配置

许多用户有直接使用域名访问 Web 网站的习惯，即在浏览器中不愿输入“www.testedu.cn”，而直接输入“tested.cn”来访问。然而，并不是所有的 Web 网站都支持这种访问方式，只有 DNS 服务器能直接解析这种域名的网站才可以。

DNS 服务器默认只能解析完全规范域名 FQDN，不能直接将域名解析成 IP 地址。为了方便用户访问，可以在 DNS 服务器的区域文件中加入下面一条特殊的 A 资源记录，以便支持实现直接解析域名功能，具体步骤如下。

(1) 在区域数据库“db.testedu.cn”文件尾部加入以下语句：

```
@       IN      A       192.168.0.110
```

(2) 重新启动 BIND 服务：

```
# service named restart
停止 named:                                              [确定]
启动 named:                                              [确定]
```

(3) 使用 host 命令进行解析：

```
# host testedu.cn
```

显示结果如下所示：

```
testedu.cn has address 192.168.0.110
```

完成以上配置后，用户在浏览器输入“tested.cn”域名时，就可以直接访问到192.168.0.110 这台主页服务器了。

6.9.5　泛域名解析配置

有些时候，用户输入域名访问某些网站的时候，可能会出现写错主机名或漏写的情况。如本应写成“www.testedu.cn”的域名写成了“wwww.testedu.cn”或者“ww.testedu.cn”造成用户无法访问网站。此时可采用泛域名解析解决这个问题。泛域名是指一个域名下的所有主机和子域名都被解析到同一个 IP 地址上。

可以在 DNS 服务器的区域文件末尾加入下面一条特殊的 A 资源记录(符号“*”是代表任何字符的通配符)，以便支持实现泛域名解析功能。具体做法如下。

(1) 在区域数据库文件 db.testedu.cn 末尾添加以下语句：

```
*.testedu.cn.   IN     A      192.168.0.120
```

或者添加：

```
*       IN     A     192.168.0.120
```

(2) 重新启动 BIND 服务：

```
# service named restart
停止 named:                                                   [确定]
启动 named:                                                   [确定]
```

(3) 使用 host 命令进行解析：

```
# host ww.testedu.cn
ww.testedu.cn has address 192.168.0.110

# host wwww.testedu.cn
wwww.testedu.cn has address 192.168.0.110

# host testaaa.testedu.cn
testaaa.testedu.cn has address 192.168.0.110
```

由上述结果可以看出，配置了泛域名解析后，对于区域数据库中不存在的域名进行解析时，其解析出来的 IP 地址都会指向泛域名解析所指定的 IP 地址。

注意： 泛域名解析只能针对域名中的任意主机名进行解析，如果用户输入的域名错误的话是不能解析成功的。例如，用户输入“www.testeducn”的话是无法解析出 IP 地址的。

6.10　辅助域名服务器配置案例

当主域名服务器的负载超过一定限额时，就应该使用辅助域名服务器，以缓解主域名服务器的压力。当主域名服务器出现死机或者故障时，辅助域名服务器还可以提供主域名服务器的功能。辅助名称服务器也可以向客户机提供域名解析功能，但它与主要名称服务器不同的是，它的数据不是直接输入的，而是从其他域名服务器(主域名服务器或其他的

辅助域名服务器)中复制过来的，所以辅助名称服务器中的数据无法被修改。

当启动辅助域名服务器时，它会和主域名服务器建立联系，并从中复制数据。在辅助域名服务器工作时，还会定期地与主域名服务器同步，保证副本与正本数据的一致性。在大型网络中，经常设置多台辅助域名服务器，具有以下优点：

- 提供容错能力。当主域名服务器发生故障时，由辅助域名服务器提供服务。
- 分担主域名服务器的负担。在 DNS 客户端较多的情况下，通过架设辅助域名服务器完成对客户端的查询服务，可以有效地减轻主域名服务器的负担。
- 加快解析的速度。对于一些大型网络，可通过在本地局域网架设辅助域名服务器的方式解决用户访问大型网络的主域名服务器缓慢的问题，减少 DNS 查询的外网通信量，加快了 DNS 解析的速度。

辅助名称服务器的主配置文件是/etc/named.conf，也需要设置服务器的选项和根区域，方法与配置主要名称服务器的方法相同。在配置辅助名称服务器时，只需要提供区域名和主要名称服务器的 IP 地址，因为一个辅助名称服务器不需要在本地建立各种资源记录，而是通过一个区域复制过程来得到主要名称服务器上的资源记录。下面通过实例来讲解设置方法。

【例 6-3】主域名服务器的域名为 testedu.cn，IP 地址为 192.168.0.104；现需要在 IP 地址为 192.168.0.103 的服务器上设置辅助域名服务器。具体操作步骤如下。

(1) 在主域名服务器上修改 BIND 主配置文件/var/named/chroot/etc/named.conf。在 option 选项中添加辅助域名服务器 IP 地址如下所示：

```
allow-transfer{192.168.0.103;};
```

(2) 在主域名服务器上修改/etc/named.rfc1912.zones 文件，修改后的区域内容如下：

```
zone "testedu.cn" IN {
       type master;
       file "db.testedu.cn";
       allow-update {none;};
};

zone "0.168.192.in-addr.arpa" IN{
       type master;
       file"rev.testedu.cn";
       allow-update {none;};
};
```

(3) 设置主域名服务器 IP、DNS 地址，重启 BIND 服务：

```
# ifconfig eth0 192.168.0.104           //设置主域名服务器 IP 地址
# service named restart                 //重启 named 服务
# echo "nameserver 192.168.0.104" >/etc/resolv.conf   //设置 DNS
```

(4) 测试主域名服务器：

```
# host -l testedu.cn                    //正向域名解析测试
testedu.cn has address 192.168.0.110
testedu.cn name server ns.testedu.cn.
*.testedu.cn has address 192.168.0.110
file.testedu.cn has address 192.168.0.113
ftp.testedu.cn has address 192.168.0.112
```

```
mail.testedu.cn has address 192.168.0.111
ns.testedu.cn has address 192.168.0.104
pop.testedu.cn has address 192.168.0.111
smtp.testedu.cn has address 192.168.0.111
www.testedu.cn has address 192.168.0.110
www.testedu.cn has address 192.168.0.120
www.testedu.cn has address 192.168.0.130

# host 192.168.0.111                    //反向域名解析测试
111.0.168.192.in-addr.arpa domain name pointer smtp.testedu.cn.
111.0.168.192.in-addr.arpa domain name pointer pop.testedu.cn.
111.0.168.192.in-addr.arpa domain name pointer mail.testedu.cn.
```

(5) 配置一台新辅助域名服务器，进入到配置文件目录，打开 named.conf 配置文件：

```
# cd /var/named/chroot/etc/          //进入到配置文件目录
# vi named.conf                      //用 vi 编辑器打开配置文件
```

修改 named.conf 主配置文件内容：(带注释语句为修改内容)

```
options {
        listen-on port 53 { 192.168.0.103; }; //修改为本地 IP 地址
        //      listen-on-v6 port 53 { ::1; };  //禁用 IPv6 监听
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        allow-query     { any; };              //将 localhost 更改为 any
        allow-query-cache { localhost; };
};
logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};
view localhost_resolver {
        match-clients     { any; };              //将 localhost 更改为 any
        match-destinations { any; };             //将 localhost 更改为 any
        recursion yes;
        include "/etc/named.rfc1912.zones";
};
```

(6) 配置辅助域名服务器，修改 named.rfc1912.zones 主配置文件。在文件末尾加入以下内容：

```
zone "testedu.cn" IN {
        type slave;
        file "slaves/db.testedu.cn";
        masters{192.168.0.104;};
};

zone "0.168.192.in-addr.arpa" IN{
        type slave;
        file"slaves/rev.testedu.cn";
        masters{192.168.0.104;};
};
```

(7) 设置辅助域名服务器 IP、DNS 地址，启动 BIND 服务：

```
# ifconfig eth0 192.168.0.103           //设置主域名服务器 IP 地址
# service named start                   //重启 named 服务
# echo "nameserver 192.168.0.103" >/etc/resolv.conf   //设置辅助域名 DNS
```

(8) 测试辅助域名服务器：

```
# host -l testedu.cn                    //正向域名解析测试
testedu.cn has address 192.168.0.110
testedu.cn name server ns.testedu.cn.
*.testedu.cn has address 192.168.0.110
file.testedu.cn has address 192.168.0.113
ftp.testedu.cn has address 192.168.0.112
mail.testedu.cn has address 192.168.0.111
ns.testedu.cn has address 192.168.0.104
pop.testedu.cn has address 192.168.0.111
smtp.testedu.cn has address 192.168.0.111
www.testedu.cn has address 192.168.0.110
www.testedu.cn has address 192.168.0.120
www.testedu.cn has address 192.168.0.130

# host 192.168.0.111                    //反向域名解析测试
111.0.168.192.in-addr.arpa domain name pointer smtp.testedu.cn.
111.0.168.192.in-addr.arpa domain name pointer pop.testedu.cn.
111.0.168.192.in-addr.arpa domain name pointer mail.testedu.cn.
```

以上结果表明辅助域名服务器测试正常，配置完成。

6.11 高速缓存域名服务器配置案例

高速缓存服务器主要用于域名缓存，无需创建区域或者区域数据库文件，只需将其启动即可。高速缓存服务器本身并不管理任何区域，而是将所有查询转发到其他 DNS 服务器处理，但是 DNS 客户端仍然可以向它请求查询。当只缓存服务器从其他 DNS 服务器收到查询结果后，除了返回给客户机外，还会将结果保存在缓存中。当下一个 DNS 客户端再查询相同的域名数据时，就可以从高速缓存中得到结果，从而加快对 DNS 客户端的响应速度。如果在局域网中建立一台这样的 DNS 服务器，就可以提高客户机 DNS 的查询效率并减少内部网络与外部网络的流量。

架设只缓存服务器非常简单，只需要建立主配置文件 named.conf 即可。下面利用实例进行具体说明。

【例 6-4】主域名服务器的域名为 testedu.cn，IP 地址为 192.168.0.104；现需要在 IP 地址为 192.168.0.105 的服务器上部署高速缓存服务器。具体操作步骤如下。

(1) 配置一台新的高速缓存服务器，修改其主配置文件 named.conf，如下所示：

```
options {
        listen-on port 53 { 192.168.0.105; };
        forwarders { 192.168.0.104;};转发到 192.168.0.104 DNS 服务器进行查询
        forward only;             ;只转发，自己不提供解析服务
//      listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
```

```
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
// Those options should be used carefully because they disable port
        // randomization
        // query-source    port 53;
        // query-source-v6 port 53;

        allow-query     { any; };
        allow-query-cache { localhost; };
};
logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};
view localhost_resolver {
        match-clients      { any; };
        match-destinations { any; };
        recursion yes;
        include "/etc/named.rfc1912.zones";
};
```

(2) 设置高速缓存域名服务器 IP、DNS 地址，启动 BIND 服务：

```
# ifconfig eth0 192.168.0.105           //设置主域名服务器 IP 地址
# service named start                   //重启 named 服务
# echo "nameserver 192.168.0.105" >/etc/resolv.conf   //设置辅助域名 DNS
```

(3) 测试高速缓存域名服务器：

```
# host -l testedu.cn                    //正向域名解析测试
testedu.cn has address 192.168.0.110
testedu.cn name server ns.testedu.cn.
*.testedu.cn has address 192.168.0.110
file.testedu.cn has address 192.168.0.113
ftp.testedu.cn has address 192.168.0.112
mail.testedu.cn has address 192.168.0.111
ns.testedu.cn has address 192.168.0.104
pop.testedu.cn has address 192.168.0.111
smtp.testedu.cn has address 192.168.0.111
www.testedu.cn has address 192.168.0.110
www.testedu.cn has address 192.168.0.120
www.testedu.cn has address 192.168.0.130

# host 192.168.0.111                    //反向域名解析测试
111.0.168.192.in-addr.arpa domain name pointer smtp.testedu.cn.
111.0.168.192.in-addr.arpa domain name pointer pop.testedu.cn.
111.0.168.192.in-addr.arpa domain name pointer mail.testedu.cn.
```

以上结果表明，高速缓存服务器测试正常，配置成功。

6.12　本 章 小 结

DNS 是 Internet 上必不可少的一种网络服务，它提供把域名解析为 IP 地址的服务，是网络中的计算机都必须要使用的服务之一。本章首先介绍了 DNS 的工作原理，然后介

绍了用 BIND 软件架设 DNS 服务器的方法，包括 BIND 的安装、运行和配置，以及 chroot、负载均衡、泛域名、辅助域名服务器、只缓存服务器等特殊功能的配置方法。最后给出了主域名服务器、辅助域名服务器、高速缓存服务器的综合实例，方便读者学习和实验。

6.13 课后习题

1. 填空题

(1) DNS 的主要作用是提供__________的转换。

(2) 使用__________命令查询 named 进程是否运行。

(3) 区域数据库文件分为__________、__________、__________三种。

2. 选择题

(1) testedu.com 公司的网络管理员，在 Linux 工作站上安装了 BIND 软件，配置实现了 DNS 服务，辅助域名服务器。在工作站上的 named.conf 文件中，BENET.COM 区域的类型是(　　)。

A. master　　B. hint　　C. slave　　D. server

(2) 应用层 DNS 协议主要用于实现(　　)网络服务功能。

A. 网络设备名字到 IP 地址的映射　　B. 网络硬件地址到 IP 地址的映射

C. 进程地址到 IP 地址的映射　　D. 用户名到进程地址的映射

(3) 测试 DNS 主要使用以下(　　)命令。

A. Ping　　B. Ipcofig　　C. nslookup　　D. Winipcfg

(4) CentOS 5 中，默认情况下 DNS 区域数据文件保存在(　　)目录中。

A. /etc/named　　B. /var/named　　C. /etc/bind　　D. /var/bind

(5) 下列(　　)命令用来在 DNS 配置文件中定义名称查询转发。

A. allowquery　　B. allowupdate　　C. forwarder　　D. forwarders

(6) 在 CentOS 5 系统中构建 BIND 服务器，并能够正确解析 www.benet.com 的 IP 地址，请问(　　)类型的 BIND 服务器需要在本机保存 benet.com 区域的数据库文件。

A. 缓存域名服务器　　B. 主域名服务器

C. 从域名服务器　　D. 转发域名服务器

(7) 下列(　　)命令可以用来进行 DNS 查询。

A. nslookup　　B. dig　　C. query　　D. host

(8) 如果要限制 DNS 查询的范围，需要在 DNS 主配置文件中加入(　　)语句。

A. allowtrans　　B. allowquery　　C. allowupdate　　D. acl

3. 简答题

(1) 简述 DNS 服务器的工作过程。

(2) 什么是域名解析？

(3) 简述 DNS 递归解析的过程。

第 7 章

Samba 服务的配置及应用

文件共享是网络环境下计算机最常用的功能之一。在第 5 章中，我们介绍了一种可在 Linux/Unix 主机之间方便共享文件的 NFS 服务。本章将介绍的 Samba 服务是一种可在 Windows、Linux、Unix 三种操作系统共享文件的服务。

7.1 Samba 服务概述

在 Linux/Unix 系统中共享文件，我们可以使用 NFS 服务来完成，客户端可以很方便地将服务器资源挂载到本地目录上面，像使用本地资源一样使用服务器资源。在 Windows/DOS 的操作系统中，微软也开发了一种实现实时共享的服务协议，叫做 Server Message Block，简称 SMB 协议。1996 年，SMB 改名成为 Common Internet File System，简称 CIFS，并提供了开源版本。就像 NFS 使用 RPC 服务作为支撑一样，CIFS 使用 IBM 的 NetBIOS 协议作为底层协议支撑。但是想在 Windows 与 Linux 系统之间共享资源却比较困难。

在早期的网络中，文件数据在 Unix、Linux、Windows(DOS)不同主机之间的共享数据大多是使用 FTP 服务器软件来进行的。但使用 FTP 传输文件却无法实现直接修改远程服务器主机上面的文件数据的功能。如果用户想要更改 Linux 主机上面的某个文件时，必须先将 FTP 服务器的资源文件下载到本地进行修改，修改完成后再将资源文件上传到服务器上。但这样的资源共享方式有很大的弊端，尤其是在多用户需同时修改同一服务器文件的时候，因为下载、修改、上传的时间差，容易出现用户修改的数据丢失的情况。

1991 年，澳大利亚国立大学的大学生 Andrew Tridgwell 拥有三种不同的操作系统主机：Microsoft 的 DOS 操作系统、DEC 公司的 Digital Unix 系统以及 Sun 的 Unix 系统。在当时，DEC 公司开发出一套称为 PATHWORKS 的软件实现了 DEC 的 Unix 与个人计算机的 DOS 两个操作系统的文件数据共享，而 Sun 的 Unix 操作系统无法通过这个软件来达到文件共享的目的。Tridgwell 通过对数据包进行分析与研究，利用软件逆向工程而开发出 Samba 这个软件。Samba 可以让 Linux/Unix 与 Windows 主机之间能彼此通过 SMB 协议实现文件共享。Windows SMB 与 Linux Samba Server 系统的关联如图 7-1 所示。

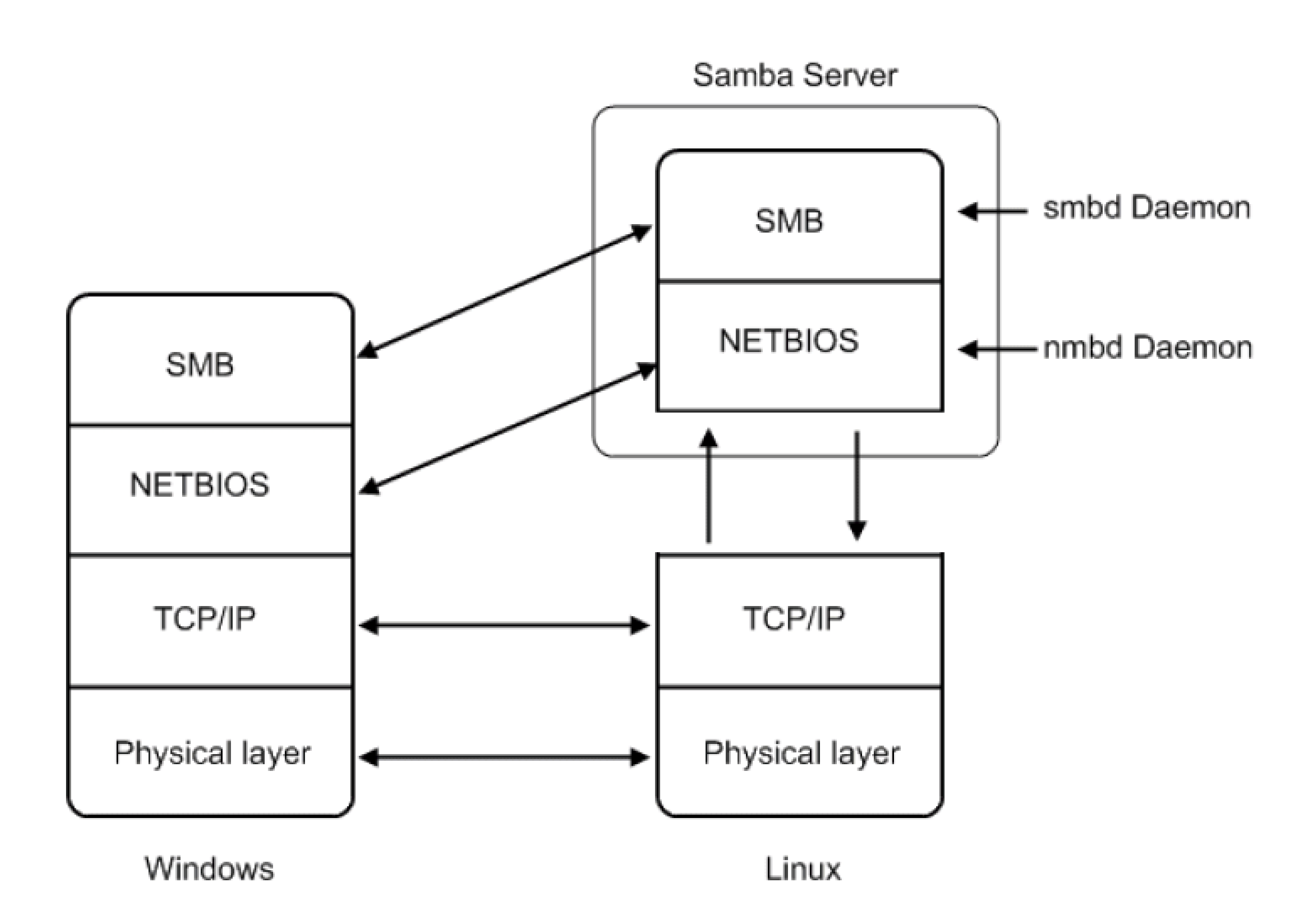


图 7-1 Windows SMB 与 Linux Samba Server 关联示意图

Samba 服务器功能的实现主要依靠处理文件和打印共享请求的 SMB 服务和处理

NetBIOS 名称服务请求和网络浏览功能的 NMBD 服务两个守护进程实现。当启动 Samba Server 时，这两个服务会一起启动。smbd 守护进程监听 TCP 的 139 和 445 端口，nmbd 进程监听 UDP 的 137 和 138 端口。这两个守护进程的配置保存在/etc/samba/smb.conf 中。smb 守护进程主要是处理到来的 smb 数据包，使的 Linux 系统用户能够访问 Windows 或 Linux 共享的资源。nmbd 守护进程使 Windows 或 Linux 客户端能够浏览服务器提供的 Samba 资源，所以两个守护进程相辅相成。

7.2　Samba 服务的安装

7.2.1　Samba 软件包介绍

与 Samba 服务有关的软件包有以下几个。

- samba*.rpm：Samba 服务器端软件。
- samba-common-*.rpm：包括 Samba 服务器端和 Samba 客户端需要的通用工具及相关的库文件。
- samba-client*.rpm：Samba 客户端软件，包括 smbclient 命令。
- system-config-samba*.rpm：Redhat 公司专门为 Samba 服务器管理编写的图形界面的管理工具，该工具是 Redhat 系统管理工具中的一部分。
- samba-swat*.rpm：安装后提供通过浏览器对 Samba 服务器进行图形化管理(Web 方式)的功能。

其中 samba-common、samba-client 默认已安装，system-config-samba 及 samba-swat 都是使用图形化方式管理 samba 的工具，用户可根据需要选择安装。

7.2.2　Samba 软件包安装

下面介绍 Samba 的安装方法。首先确认 Samba 服务有没有正确安装可以运行如下命令：

```
# rpm -qa | grep samba
```

输出结果如下：

```
# rpm -qa | grep samba
samba-common-3.0.33-3.28.el5
samba-client-3.0.33-3.28.el5
```

已经安装的软件包里面没有包含 samba 的服务器端软件。用户可将光盘加载，并进入 CentOS 5 的光盘 rpm 目录/media/CentOS*/CentOS/，并执行如下命令：

```
#rpm -ivh samba-* system-config-samba-* --force
```

显示结果如下：

```
#rpm -ivh samba-* system-config-samba-* --force
Preparing...                ##################################### [100%]
1:samba-common            ##################################### [ 20%]
2:samba                  ######################################## [ 40%]
```

```
3:samba-client          ######################################## [ 60%]
4:samba-swat            ######################################## [ 80%]
5:system-config-samba   ######################################## [100%]
```

注意： 如果在运行 rpm 命令安装时提示 warning: samba-3.0.33-3.28.el5.i386.rpm: Header V3 DSA signature: NOKEY, key ID e8562897，说明 rpm 在安装软件包时没有找到合适的钥匙来校验签名。这可能是由于通过 yum 安装了旧版本的 GPG keys 造成的，解决办法就是运行#rpm --import /etc/pki/rpm-gpg/RPM*，这样再次安装软件时就不会有这个错误提示了。

再次运行 rpm -qa | grep samba 命令，可以得到如下结果：

```
# rpm -qa | grep samba
samba-client-3.0.33-3.28.el5
samba-common-3.0.33-3.28.el5
system-config-samba-1.2.41-5.el5
samba-3.0.33-3.28.el5
samba-swat-3.0.33-3.28.el5
```

至此，Samba 服务器软件包已经全部安装成功。

7.3 Samba 服务的配置文件

Samba 服务的主配置文件是/etc/samba/smb.conf。除了主配置文件，Samba 服务器还包含一些其他的配置文件，如：/etc/samba/smbpasswd 是 Samba 服务的用户密码文件；/etcsamba/smbusers 是 Samba 服务的用户文件；/var/log/samba 是 Samba 服务的日志文件；/etc/init.d/smb 是 Samba 服务的启动脚本文件。

注意： 软件版本不同，配置文件所在的目录也可能不同。我们可以通过查询软件包的文件结构(rpm –qa 软件包名称)来确定各个文件的位置。如查询 Samba 的软件包文件结构可运行# rpm -ql samba。

7.3.1 Samba 的主配置文件

/etc/samba/smb.conf 是 Samba 服务的主配置文件。Samba 服务器功能非常丰富，有很多功能在 Windows 下都无法实现，主配置文件中内容也非常庞大。smb.conf 配置文件一共分为两个部分，第一部分为全局参数设置，第二部分为共享参数设置。其中第二部分共享参数设置分为两个子版块：第一个子版块为[homes]共享，主要设置 Samba 服务器目录共享功能；第二个子版块为[printers]共享，主要设置 Samba 服务器打印机共享功能。

smb.conf 文件中已经包含了许多注释和示例。其中每行以“#”开头的语句是配置文件的说明语句，每行以“;”开头的是配置文件中可选的配置选项或示例，默认是不起作用的。用户如需要配置，直接去掉每行前面的“;”，并将选项的内容替换为个人设定的内容即可。在配置参数时有以下几个共性。

(1) 对参数进行配置时基本都采用“参数 = 值”的方式，如参数有多个值时，多个值之间用空格分隔。

(2) 当可以使用用户和组作为参数值时，值为组时需在组名前加@。

(3) 以 “;” 或 “#” 开始的是注释行(在执行时将被忽略)。

(4) 方括号标识表示为标志，比如[global]为全局配置标识。

(5) 一般当全局配置与某个共享资源配置发生冲突时，共享资源配置优先。

(6) 关键字对大小写不敏感。

因主配置文件较长，使用如下命令可以过滤掉注释字符，查看初始主配置文件内容：

```
#  grep -v -E "^#|^$|^;" /etc/samba/smb.conf
[global]
        workgroup = MYGROUP
        server string = Samba Server Version %v
        security = user
        passdb backend = tdbsam
        cups options = raw
        username map = /etc/samba/smbusers
[homes]
        comment = Home Directories
        browseable = no
        writable = yes
[printers]
        comment = All Printers
        path = /var/spool/samba
        browseable = no
        printable = yes
```

主配置文件的具体配置方法将在 7.4 节中做详细介绍。

7.3.2　Samba 的用户密码文件

/etc/samba/smbpasswd 是 Samba 服务的用户密码文件。初始安装时该文件不存在，需要用户使用时创建。smbpasswd 文件保存的是能访问 Samba 服务器资源的用户名与密码。Samba 服务器的用户必须是 Linux 系统中已经注册存在的用户，即 smbpasswd 文件中的用户名必须是系统的/etc/passwd 文件中的用户名的子集，只有在 Linux 系统中存在的用户才能添加到 Samba 服务器中。当 Linux 系统中用户名密码被删除时，Samba 服务器中所对应的用户名密码也将被删除。

7.3.3　Samba 用户对应文件

/etc/samba/smbusers 是 Samba 服务的用户对应文件，主要记录访问 Samba 服务器共享资源的用户转换及对应关系。比如 Windows 操作系统的超级管理员账户名为 administrator，而 Linux 系统的超级用户名字为 root，如果需要双方系统使用超级管理员用户进行共享资源访问，并确定 Windows 超级管理员与 Linux 超级用户有同等权限，则可以在文件中设置对应管理。这样 Windows 的超级管理员登录到 Samba 服务器后便实际上对应了 Linux 的 root 用户，连接后 administrator 身份将转换成 root 身份。一般出于安全性考虑，不建议用户使用此配置文件。

7.3.4 Samba 日志文件

/var/log/samba 目录记录 Samba 服务器的 smbd 和 nmbd 守护进程的日子文件内容。Smbd 将日志记录到/var/log/samba/smdb.log，nmbd 将日志记录到/var/log/samba/nmbd.log 中，同时记录有哪些主机在什么时间访问了 Samba 服务器。

可以通过以下命令查看 Samba 服务的日志文件：

```
ls -l /var/log/samba/
```

显示结果如下所示：

```
# ls -l /var/log/samba/
总计 32
drwx------ 4 root root 4096 02-15 23:45 cores
-rw-r--r-- 1 root root    0 03-23 01:57 nmbd.log
-rw-r--r-- 1 root root 2264 03-23 01:57 nmbd.log.1
-rw-r--r-- 1 root root    0 03-23 01:57 smbd.log
-rw-r--r-- 1 root root 3521 03-23 01:57 smbd.log.1
```

7.3.5 Samba 服务的启动脚本文件

Samba 服务的启动脚本文件是/etc/init.d/smb。我们可以直接运行这个脚本来启动或停止 Samba 服务，也可以根据需要修改脚本文件内容。

7.4 Samba 服务器的配置

Samba 服务器的配置主要是修改主配置文件/etc/samba/smb.conf 的内容。主配置文件由全局参数、共享参数两个部分组成。本节首先对 Samba 服务器配置步骤进行介绍，然后对主配置文件中各项参数做详细介绍。

7.4.1 Samba 服务器配置步骤

Samba 服务器的配置步骤非常简单，如下所示：

(1) 检查 Samba 服务软件包是否安装，如未安装则需要先安装 samba 服务软件包。

(2) 根据需要创建共享目录。

(3) 修改 smb.conf 主配置文件中相应参数。

(4) 创建 Samba 访问用户，如服务器只开启匿名访问，则此步骤跳过。

(5) 启动 Samba 服务器。

(6) 编辑防火墙规则或者关闭防火墙。并使 SELinux 允许 Samba 服务运行。

(7) 在客户端对 Samba 服务器进行测试。

7.4.2 Samba 全局参数

全局参数用于[global]的<section>选项定义中，用于说明 samba 服务器的一些基本

属性。

```
[global]

config file = /usr/local/samba/lib/smb.conf.%m
```

#config file 可以让你使用另一个配置文件来覆盖缺省的配置文件。如果文件不存在，则该项无效。这个参数很有用，可以使得 samba 配置更灵活，可以让一台 samba 服务器模拟多台不同配置的服务器。比如，你想让 PC1(主机名)这台电脑在访问 Samba Server 时使用它自己的配置文件，那么先在/etc/samba/host/下为 PC1 配置一个名为 smb.conf.pc1 的文件，然后在 smb.conf 中加入：config file = /etc/samba/host/smb.conf.%m。这样当 PC1 请求连接 Samba Server 时，smb.conf.%m 就被替换成 smb.conf.pc1。这样，对于 PC1 来说，它所使用的 Samba 服务就是由 smb.conf.pc1 定义的，而其他机器访问 Samba Server 则还是应用 smb.conf。

workgroup = WORKGROUP
#设定 Samba Server 所要加入的工作组或者域。

server string = Samba Server Version %v
#设定 Samba Server 的注释，可以是任何字符串，也可以不填。变量%v 表示显示 Samba 的版本号。

netbios name = smbserver
#设置 Samba Server 的 NetBIOS 名称。如果不填，则默认会使用该服务器的 DNS 名称的第一部分。netbios name 和 workgroup 名字不要设置成一样了。

interfaces = lo eth0 192.168.12.2/24 192.168.13.2/24
#设置 Samba Server 监听哪些网卡，可以写网卡名，也可以写该网卡的 IP 地址。

hosts allow =IP 地址
#表示允许连接到 Samba Server 的客户端，多个参数以空格隔开。可以用一个 IP 表示，也可以用一个网段表示。hosts deny 与 hosts allow 刚好相反。
#例如：hosts allow=172.17.2.EXCEPT172.17.2.50
#表示容许来自 172.17.2.*.*的主机连接，但排除 172.17.2.50
#hosts allow=172.17.2.0/255.255.0.0
#表示容许来自 172.17.2.0/255.255.0.0 子网中的所有主机连接
#hosts allow=M1，M2
#表示容许来自 M1 和 M2 两台计算机连接
#hosts allow=@xq
#表示容许来自 XQ 网域的所有计算机连接

max connections = 0
#max connections 用来指定连接 Samba Server 的最大连接数目。如果超出连接数目，则新的连接请求将被拒绝。0 表示不限制。

deadtime = 0
#deadtime 用来设置断掉一个没有打开任何文件的连接时间。单位是分钟，0 代表 Samba Server 不自动切断任何连接。

time server = yes/no
#time server 用来设置让 nmdb 成为 windows 客户端的时间服务器。

log file = /var/log/samba/log.%m
#设置 Samba Server 日志文件的存储位置以及日志文件名称。在文件名后加个变量%m(主机名)，表示对每台访问 Samba Server 的机器都单独记录一个日志文件。如果 pc1、pc2 访问过 Samba Server，就会在/var/log/samba 目录下留下 log.pc1 和 log.pc2 两个日志文件。

max log size = 50
#设置 Samba Server 日志文件的最大容量，单位为 kB，0 代表不限制。

security = user
#设置用户访问 Samba Server 的验证方式，一共有五种验证方式。
#1. share：用户访问 Samba Server 不需要提供用户名和口令，安全性能较低。
#2. user：Samba Server 共享目录只能被授权的用户访问，由 Samba Server 负责检查账号和密码的正确性。账号和密码要在本 Samba Server 中建立。
#3. server：依靠其他 Windows NT/2000 或 Samba Server 来验证用户的账号和密码，是一种代理验证。此种安全模式下，系统管理员可以把所有的 Windows 用户和口令集中到一个 NT 系统上，使用 Windows NT 进行 Samba 认证，远程服务器可以自动认证全部用户和口令，如果认证失败，Samba 将使用用户级安全模式作为替代的方式。
#4. domain：域安全级别，使用主域控制器（PDC）来完成认证。
#5. ADS：Samba 自 3.0 开始可以完美支持 Windows 的活动目录。采用该模式的客户端必须是 Windows2000 以上版本。

passdb backend = tdbsam
#passdb backend 就是用户后台的意思。目前有三种后台：smbpasswd、tdbsam 和 ldapsam。sam 应该是 security account manager（安全账户管理）的简写。
#1.smbpasswd：该方式是使用 smb 自己的工具 smbpasswd 来给系统用户（真实用户或者虚拟用户）设置一个 Samba 密码，客户端就用这个密码来访问 Samba 的资源。smbpasswd 文件默认在 /etc/samba 目录下，不过有时候要手工建立该文件。将改语句替换成 smb passwd file = /etc/samba/smbpasswd 即可转或成 smbpasswd 模式。
#2.tdbsam：该方式则是使用一个数据库文件来建立用户数据库。数据库文件叫 passdb.tdb，默认在/etc/samba 目录下。passdb.tdb 用户数据库可以使用 smbpasswd -a 来建立 Samba 用户，不过要建立的 Samba 用户必须先是系统用户。我们也可以使用 pdbedit 命令来建立 Samba 账户。pdbedit 命令的参数很多，我们列出几个主要的。
#pdbedit -a username：新建 Samba 账户。
#pdbedit -x username：删除 Samba 账户。
#pdbedit -L：列出 Samba 用户列表，读取 passdb.tdb 数据库文件。
#pdbedit -Lv：列出 Samba 用户列表的详细信息。
#pdbedit -c “[D]” -u username：暂停该 Samba 用户的账号。
#pdbedit -c “[]” -u username：恢复该 Samba 用户的账号。
#3.ldapsam：该方式则是基于 LDAP 的账户管理方式来验证用户。首先要建立 LDAP 服务，然后设置“passdb backend = ldapsam:ldap://LDAP Server”

encrypt passwords = yes/no
#是否将认证密码加密。因为现在 windows 操作系统都是使用加密密码，所以一般要开启此项。不过配置文件默认已开启。

smb passwd file = /etc/samba/smbpasswd
#用来定义 samba 用户的密码文件。smbpasswd 文件如果没有那就要手工新建。

username map = /etc/samba/smbusers
#用来定义用户名映射，比如可以将 root 换成 administrator、admin 等。不过要事先在 smbusers 文件中定义好。比如：root = administrator admin，这样就可以用 administrator 或 admin 这两个用户来代替 root 登陆 Samba Server，更贴近 windows 用户的习惯。

guest account = nobody
#用来设置 guest 用户名。

socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192

#用来设置服务器和客户端之间会话的 Socket 选项，可以优化传输速度。

domain master = yes/no
#设置 Samba 服务器是否要成为网域主浏览器，网域主浏览器可以管理跨子网域的浏览服务。

local master = yes/no
#local master 用来指定 Samba Server 是否试图成为本地网域主浏览器。如果设为 no，则永远不会成为本地网域主浏览器。但是即使设置为 yes，也不等于该 Samba Server 就能成为主浏览器，还需要参加选举。

preferred master = yes/no
#设置 Samba Server 一开机就强迫进行主浏览器选举，可以提高 Samba Server 成为本地网域主浏览器的机会。如果该参数指定为 yes 时，最好把 domain master 也指定为 yes。使用该参数时要注意：如果在本 Samba Server 所在的子网有其他的机器(不论是 windows NT 还是其他 Samba Server)也指定为首要主浏览器时，那么这些机器将会因为争夺主浏览器而在网络上大发广播，影响网络性能。
如果同一个区域内有多台 Samba Server，将上面三个参数设定在一台即可。

os level = N
#设置 samba 服务器的 os level。N 是整数。该参数决定 Samba Server 是否有机会成为本地网域的主浏览器。os level 从 0 到 255，winNT 的 os level 是 32，win95/98 的 os level 是 1。Windows 2000 的 os level 是 64。如果设置为 0，则意味着 Samba Server 将失去浏览选择。如果想让 Samba Server 成为 PDC，那么将它的 os level 值设大些。

domain logons = yes/no
#设置 Samba Server 是否要作为本地域控制器。主域控制器和备份域控制器都需要开启此项。

logon . = %u.bat
#当使用者用 windows 客户端登录，那么 Samba 将提供一个登录档。如果设置成%u.bat，那么就要为每个用户提供一个登录档。如果人比较多，那就比较麻烦。可以设置成一个具体的文件名，比如 start.bat，那么用户登录后都会去执行 start.bat，而不用为每个用户设定一个登录档了。这个文件要放置在[netlogon]的 path 设置的目录路径下。

wins support = yes/no
#设置 samba 服务器是否提供 wins 服务。

wins server = wins 服务器 IP 地址
#设置 Samba Server 是否使用别的 wins 服务器提供 wins 服务。

wins proxy = yes/no
#设置 Samba Server 是否开启 wins 代理服务。

dns proxy = yes/no
#设置 Samba Server 是否开启 dns 代理服务。

load printers = yes/no
#设置是否在启动 Samba 时就共享打印机。

printcap name = cups
#设置共享打印机的配置文件。

printing = cups
#设置 Samba 共享打印机的类型。现在支持的打印系统有：bsd, sysv, plp, lprng, aix, hpux, qnx

7.4.3 Samba 共享参数

共享参数定义了除了[global]外的各个<section>中的参数。其定义了共享服务的属性。

comment = 任意字符串
#comment 是对该共享的描述，可以是任意字符串。

path = 共享目录路径
#path 用来指定共享目录的路径。可以用%u、%m 这样的变量来代替路径里的 unix 用户和客户机的 Netbios 名，用变量表示主要用于[homes]共享域。例如：如果我们不打算用 home 段作为客户的共享，而是在/home/share/下为每个 Linux 用户以他的用户名建个目录，作为他的共享目录，这样 path 就可以写成：path = /home/share/%u; 。用户在连接到这共享时具体的路径会被他的用户名代替，要注意这个用户名路径一定要存在，否则，客户机在访问时会找不到网络路径。同样，如果我们不是以用户来划分目录，而是以客户机来划分目录，为网络上每台可以访问 samba 的机器都各自建个以它的 netbios 名的路径，作为不同机器的共享资源，就可以这样写：path = /home/share/%m 。

browseable = yes/no
#browseable 用来指定该共享是否可以浏览。

writable = yes/no
#writable 用来指定该共享路径是否可写。

available = yes/no
#available 用来指定该共享资源是否可用。

admin users = 该共享的管理者
#admin users 用来指定该共享的管理员(对该共享具有完全控制权限)。在 samba 3.0 中，如果用户验证方式设置成“security=share”时，此项无效。
例如：admin users =bobyuan，jane(多个用户中间用逗号隔开)。

valid users = 允许访问该共享的用户
#valid users 用来指定允许访问该共享资源的用户。
例如：valid users = bobyuan，@bob，@tech(多个用户或者组中间用逗号隔开，如果要加入一个组就用“@+组名”表示。)

invalid users = 禁止访问该共享的用户
#invalid users 用来指定不允许访问该共享资源的用户。
例如：invalid users = root，@bob(多个用户或者组中间用逗号隔开。)

write list = 允许写入该共享目录的用户
#write list 用来指定可以在该共享下写入文件的用户。
例如：write list = bobyuan，@bob

public = yes/no
#public 用来指定该共享是否允许 guest 账户访问。

guest ok = yes/no
#意义同“public”。

除了 homes 和 printers 外，用户还可以在 smb.conf 文件中自定义共享服务名。

7.4.4　Samba 自定义变量

在共享服务的定义中，我们通过一些选项来定义共享服务的属性。在选项的定义中，我们可以使用一些 samba 预定义的变量来设置动态的选项值。下面列出几个常用的预定义变量：

- %S：当前服务名。
- %P：当前服务的根目录。
- %u：当前服务的用户名。
- %U：当前会话的用户名。
- %g：当前服务用户所在的主工作组。
- %G：当前会话用户所在的主工作组。
- %H：当前服务的用户的 Home 目录。
- %V：samba 的版本号。
- %h：运行 samba 服务机器的主机名。
- %M：客户端的主机名。
- %m：客户端的 NetBIOS 名称。
- %L：服务器的 NetBIOS 名称。
- %R：所采用的协议等级(CORE/COREPLUS/LANMAN1/LANMAN2/NT1)。
- %d：当前服务进程的 ID。
- %I：客户端的 IP。
- %T：当前日期和时间。

7.5　Samba 服务的启动与停止

7.5.1　Samba 服务的启动

启动 Samba 服务，会同时启动 smbd 和 nmbd 两个进程。我们可以使用如下命令启动 Samba 服务：

```
service smb start
```

显示结果如下所示：

```
# service smb start
启动 SMB 服务：                                        [确定]
启动 NMB 服务：                                        [确定]
```

由显示结果可知， Samba 服务已经启动成功。

用户还可以使用下面的命令启动 Samba 服务：

```
# /etc/init.d/smb start
```

显示同样的结果：

```
# /etc/init.d/smb start
启动 SMB 服务:                                              [确定]
启动 NMB 服务:                                              [确定]
```

这两种启动方式效果是一样的。其中/etc/init.d/smb start 方法适用于任何的 Linux 操作系统，而 service smb start 仅仅适用于 Red Hat 系列或与其类似的操作系统。

7.5.2 Samba 服务的停止

停止 Samba 服务也有两种方式：

```
# service smb stop
```

显示结果如下：

```
# service smb stop
关闭 SMB 服务:                                              [确定]
关闭 NMB 服务:                                              [确定]
```

或者采用如下命令：

```
#  /etc/init.d/smb stop
```

显示结果如下：

```
#  /etc/init.d/smb stop
关闭 SMB 服务:                                              [确定]
关闭 NMB 服务:                                              [确定]
```

7.5.3 设置 Samba 服务开机自运行

可以使用 chkconfig 命令设置 Samba 服务开机自运行：

```
# chkconfig --level 345 smb on
```

也可以通过 ntsysv 图像界面设置 Samba 服务的开机自运行：

```
#ntsysv --level 345
```

在弹出的 ntsysv 图形窗口中用“*”选中 smb 服务，使其能够在在开机自动启动，并选中“确定”按钮，退出 ntsysv 设置，如图 7-2 所示。

图 7-2 ntsysv 设置 smb 服务开机自启动

7.5.4 检测 Samba 服务是否正常启动

可以通过端口状态检测命令 netstat 和进程显示命令 ps 两种方式检测 Samba 服务是否正常启动：

```
#netstat -antup | grep smb
# ps -eaf | grep smbd
```

显示结果分别如下所示：

```
#netstat -antup | grep smb
tcp        0      0 0.0.0.0:139       0.0.0.0:*         LISTEN      18872/smbd
```

```
tcp        0      0 0.0.0.0:445         0.0.0.0:*       LISTEN      18872/smbd

# ps -eaf | grep smbd
root     18872     1  0 09:56 ?        00:00:00 smbd -D
root     18875 18872  0 09:56 ?        00:00:00 smbd -D
root     18980  4794  0 09:58 pts/1    00:00:00 grep smbd
```

7.5.5　修改 SELinux 状态

在 Samba 服务器中并不是所有的操作都会受到 SELinux 的影响，Samba 所涉及的 SELinux 配置参数比较多，不过在 smb.conf 文件中通过注释的方式已对其中一部分进行了说明。

下面列出的是所有与 Samba 服务器有关的 SELinux 配置。

(1) setsebool -P samba_domain_controller on

(2) setsebool -P samba_enable_home_dirs on　//如果不需在 Samba 使用默认共享的用户目录，可不必执行。

(3) setsebool -P samba_export_all_ro on　//如果在整个 Samba 所有的共享目录均为只读时运行。

(4) setsebool -P samba_export_all_rw on　//如果在整个 Samba 所有的共享目录有可写时运行。

(5) chcon -R -t samba_share_t 欲共享的本地目录名　//如果在整个 Samba 所有共享目录中有允许建立目录时运行，在执行 chcon 命令时一定要小心，如果修改了根目录安全上下文 RHEL5.x 在重新启动后将无法登录。

(6) setsebool -P smbd_disable_trans=1

chcon -R -t samba_share_t /bin/mount

chcon -R -t samba_share_t /bin/umount

//以上语句如果需要共享光驱时使用运行。

(7) setsebool -P swat_disable_trans=1　//如果需要使用时 SWAT 运行。

(8) setsebool -P smbd_disable_trans=1　//如果需要使用 PAM 模块进行访问控制时运行。

如服务器只是作为测试使用，用户也可以使用如下命令将 SELinux 设置为 permissive 模式(重启后 SELinux 恢复)：

```
# setenforce 0
```

如需要永久禁用 SELinux，修改/etc/selinux/config 文件中的 SELINUX=" disabled "，然后重启即可。

7.5.6　修改 Iptables 防火墙状态

为 Samba 服务器进行简单测试时，用户可以选择关闭 Linux 防火墙：

```
# service iptables stop
```

或者可以使用如下脚本来打开 Linux 系统防火墙。在这个脚本中，用户只需要在前 4 行设

置好服务器 IP、子网掩码、网络地址、广播地址这 4 个网络参数，再将脚本复制到终端命令行执行即可。

```
SERVER="192.168.0.1" # 服务器 IP 地址
NETMASK="255.255.0.0" # 子网掩码
NETWORK="192.168.0.0" # 网络地址
BROADCAST="192.168.255.255" # 广播地址

iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A INPUT -p udp -s ${NETWORK}/${NETMASK} -d ${SERVER}/32 -m
multiport --dports 137,138 -j ACCEPT
 iptables -A INPUT -p tcp -s ${NETWORK}/${NETMASK} -d ${SERVER}/32 -m
multiport --dports 139,445 -j ACCEPT
 iptables -A INPUT -p udp -s ${NETWORK}/${NETMASK} -d ${BROADCAST}/32 --
dport 137 -j ACCEPT
 iptables -A INPUT -p udp -d ${SERVER}/32 -m multiport --dports 137,138
-j DROP
iptables -A INPUT -p tcp -d ${SERVER}/32 -m multiport --dports 139,445 -
j DROP
iptables -A OUTPUT -s ${SERVER}/32 -d ${NETWORK}/32 -m state --state
ESTABLISHED,RELATED -j ACCEPT
```

关于 Linux 防火墙详细配置，请参阅第 11 章。

7.5.7 使用图形化方式设置 Samba 服务启动

CentOS 5 的 GNOME 图形界面中选择“系统”→“管理”→“服务器设置”→“服务”，打开“服务配置”窗口，如图 7-3 所示。在该窗口中选中 smb 服务，然后单击“开始”、“停止”、“重启”等按钮即可实现对 smb 服务的启动、停止和重启操作。在窗口中选中 smb 服务前的复选框，还可实现 NFS 服务的开机自动运行。

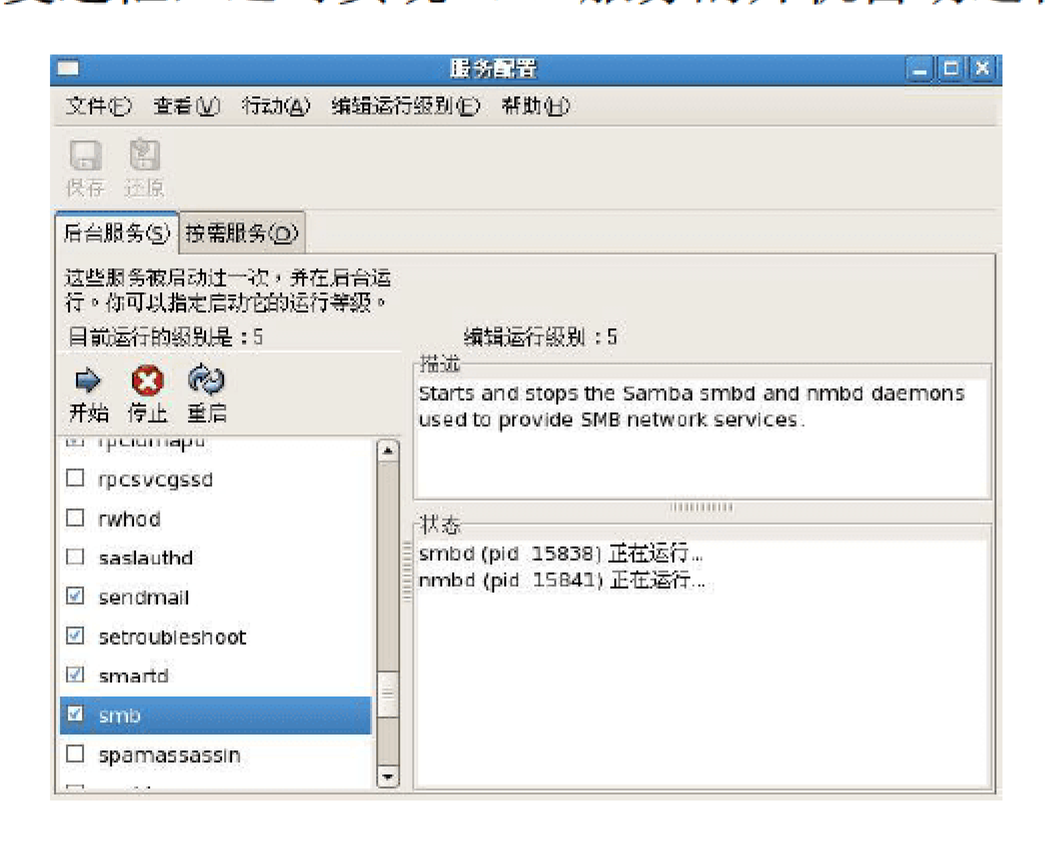


图 7-3 CentOS 5 中的“服务配置”窗口

7.6 Samba 常用工具命令

Samba 服务提供了很多的工具命令以满足不同需求，如配置文件语法检测命令 testparm，共享目录详细列表命令 smbclient，Samba 客户端访问工具命令 smbmount，

Samba 用户管理命令 smbpasswd，Samba 服务运行状态检测命令 smbstaus 命令等，下面向大家逐一介绍。

在学习命令之前，让我们先建立一个测试 Samba 服务器，并对其进行如下修改：

```
# useradd alice                    //为系统添加一个 alice 用户
# mkdir /public                    //为系统创建一个 public 目录用于 Samba 共享
# chown alice:alice /public        //将/public 文件夹的 owner 改为 alice
# service iptables stop            //关闭防火墙
# setenforce 0                     //设置 SELinux 至 permissive 模式
```

修改 Samba 配置文件/etc/samba/smb.conf，并在文件尾部添加内容如下：

```
[public]
        comment = public
        path = /public
        writeable = yes
;       browseable = yes
        valid users = alice
```

7.6.1　smbpasswd 命令

smbpasswd 命令用于在 Samba 服务器中创建用户及更改密码。使用参数如下所示：

- smbpasswd -a 增加用户(增加的用户必须在系统用户中已经存在)。
- smbpasswd -d 停用用户，禁止用户使用 Samba。
- smbpasswd -e 启用用户，让停用的用户可以使用 Samba。
- smbpasswd -n 把用户的密码设置成空(要在 global 中写入 null passwords –true)。
- smbpasswd -x 删除用户。

下面，将 alice 用户加入到 Samba 中，运行命令如下所示：

```
# smbpasswd -a alice
New SMB password:
Retype new SMB password:
```

添加 Samba 用户时需要输入 Samba 的密码，该密码可以与系统密码不同。

7.6.2　testparm 命令

testparm 命令是用来检测 Samba 服务的配置文件 smb.conf 是否存在错误。在我们配置好 smb.conf 文件后，为避免发生拼写或语法错误应使用 testparm 命令对配置文件进行检查。输入 samba 主配置文件的检测命令：

```
# testparm
```

显示结果如下所示：

```
#  testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[public]"
Loaded services file OK.
Server role: ROLE_STANDALONE
```

```
Press enter to see a dump of your service definitions

[global]
        workgroup = MYGROUP
        server string = Samba Server Version %v
        passdb backend = tdbsam
        username map = /etc/samba/smbusers
        cups options = raw

[homes]
        comment = Home Directories
        read only = No
        browseable = No

[printers]
        comment = All Printers
        path = /var/spool/samba
        printable = Yes
        browseable = No

[public]
        comment = public
        path = /public
        valid users = alice
        read only = No
```

以上结果表明 Samba 主配置文件没有问题，在回车后显示 smb.conf 全局参数和共享目录中共享参数的情况。Samba 服务默认共享了[homes]、[printers]和[public]三个目录。

7.6.3　smbclient 命令

smbclient 是用于在 Linux 或者 UNIX 系统下访问 Samba 服务器共享资源的客户端命令。简单的格式如下：

```
smbclient [-L] //ServerIP/samba 共享名称 [-U username[password]]
```

其中-L 参数表示列出 Samba 服务器的共享目录；-U 参数表示使用哪个用户访问 Samba 服务器；用户名后面可加用户登录密码。

例如用户 alice 需要访问 192.168.0.103 Samba 服务器的 public 共享，假设服务器已经配置完成，则可以使用以下命令：

```
# smbclient -L //192.168.0.103/public -U alice
Password:
Domain=[BOGON] OS=[Unix] Server=[Samba 3.0.33-3.28.el5]

        Sharename       Type      Comment
        ---------       ----      -------
        public        Disk     public
        IPC$          IPC       IPC Service (Samba Server Version 3.0.33-3.28.el5)
        Microsoft_XPS_Document_Writer:1 Printer   Microsoft XPS Document Writer
        Microsoft_Office_Document_Image_Writer:2 Printer   Microsoft
Office Document Image Writer
        Fax:3            Printer    Fax
        alice            Disk      Home Directories
Domain=[BOGON] OS=[Unix] Server=[Samba 3.0.33-3.28.el5]
```

```
    Server           Comment
    ---------        -------

    Workgroup        Master
    ---------        -------
    MYGROUP
```

7.6.4　mount 命令

mount 命令主要用于挂载 Samba 服务器共享目录到本地目录。例如需要将 192.168.0.105 Samba 服务器的 public 目录挂载到本地的/mnt/public 目录，可使用如下命令实现：

```
# mkdir /mnt/public
# mount  -t cifs -o username=alice,password=alice //192.168.0.103/public
/mnt/public

//查看目录加载情况
# mount
/dev/mapper/VolGroup00-LogVol00 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda1 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
none on /proc/fs/vmblock/mountPoint type vmblock (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/hdc on /media/CentOS_5.5_Final type iso9660
(ro,noexec,nosuid,nodev,uid=0)
//192.168.0.103/public on /mnt/public type cifs (rw,mand)
```

由上面结果可知，Samba 服务器的 public 目录已经成功加载到客户端/mnt/public 目录下。

注意： 原有的 smbmount 命令在 CentOS 5.5 版本中不再支持。mount –t smbfs 在 CentOS5.5 中也不再支持，应改用 mount –t cifs 加载 Samba 共享目录。

7.6.5　smbstatus 命令

smbstatus 命令用于查看 Samba 服务器的状态，如显示哪些主机连接到 Samba 服务器，哪些用户正在访问 Samba 服务器的文件等。

```
# smbstatus

Samba version 3.0.33-3.28.el5
PID     Username      Group         Machine
-------------------------------------------------------------------
 5610   alice         alice         sd--20110325cwb (192.168.0.101)
 5716   alice         alice         192.168.0.103 (192.168.0.103)

Service      pid     machine       Connected at
-------------------------------------------------------
public       5716   192.168.0.103  Thu Mar 20 13:39:56 2012
```

```
alice        5610   sd--20110325cwb  Thu Mar 20 13:36:06 2012
IPC$         5610   sd--20110325cwb  Thu Mar 20 13:35:16 2012

Locked files:
Pid          Uid        DenyMode   Access      R/W        Oplock
SharePath    Name    Time
------------------------------------------------------------------------
5610         502        DENY_NONE  0x100001    RDONLY     NONE
/home/alice   .    Thu Mar 20 13:36:06 2012
```

由以上结果可知，目前共有两台客户端连接至 Samba 服务器，IP 地址分别为 192.168.0.101 与 192.168.0.103，分别连接到服务器的 public、alice 与 IPC$目录。连接进程分别为 5610 和 5716。若管理员需要断开该连接，可以使用 kill 命令结束该进程。

注意： IPC$目录是 Samba 服务器为兼容 Windows 客户端而自带的一个目录，一般不需要对其进行设置。

7.6.6 smbtree 命令

smbtree 命令用于查找 Samba 工作组中的所有主机的共享目录，如工作组中由多少台 Samba 服务器，共享了哪些目录。

```
#smbtree
Password:
MYGROUP
       \\BOGON                       Samba Server Version 3.0.33-3.28.el5
```

以上结果表明，smbtree 找到一台 Samba 服务器。smbtree 的命令参数如下所示：

-D：只显示工作组名，不显示主机。

-b：使用广播查询列表，不用 WINS 服务器。

7.6.7 smbtar 命令

smbtar 命令主要功能是将 Samba 服务器的共享文件备份到某个服务器或者本地。如 Samba 出现问题，可以通过 smbtar 命令将备份文件恢复至 Samba 服务器。smbtar 命令格式可以通过/usr/bin/smbtar 脚本得到。若不允许匿名用户备份 Samba 数据，可以将/usr/bin/smbtar 文件中的-N 参数删除。

smbtar 的命令参数：

- -s：Samba 服务器 IP 地址或主机名。
- -u：备份或者恢复时所用的用户名。
- -p：备份或恢复时所使用的用户的密码。
- -x：备份或恢复的共享目录。
- -t：备份到哪个文件或设备。
- -r：从哪个文件或设备恢复数据。

例如利用 alice 用户备份 Samba 服务器 192.168.0.103 的 public 目录所有文件，执行命令如下所示：

```
# smbtar -s 192.168.0.103 -u alice -p alice -x public -t
/home/alice/public.tar
```

7.7　Samba 服务器端的配置

通过字符方式配置 Samba 服务器需要记住大量的命令和配置选项，在带有 X-Windows 图形界面操作系统的 CentOS 中，我们可以采用 system-config-samba 对 Samba 服务器进行配置。

(1) 确认是否已安装 system-config-samba 软件包。使用 rpm -qa 命令查询是否已经安装该软件包，如未安装，需进入到光盘的软件包所在目录使用 rpm -ivh system-config-samba* 命令进行安装。

```
# rpm -qa | grep system-config-samba
```

显示结果如下：

```
# rpm -qa | grep system-config-samba
system-config-samba-1.2.41-5.el5
```

说明 Samba 的配置图形工具已经成功安装。

(2) 打开 Samba 图形配置界面。我们可以直接在终端命令行输入“system-config-samba”命令进入到 Samba 服务器配置界面，也可以进入到 X-Windows 界面中，选择“系统”→“管理”→“服务器设置”→Samba，进入“Samba 服务器配置”界面，如图 7-4、图 7-5 所示。




图 7-4　启动 Samba 服务器图形配置界面

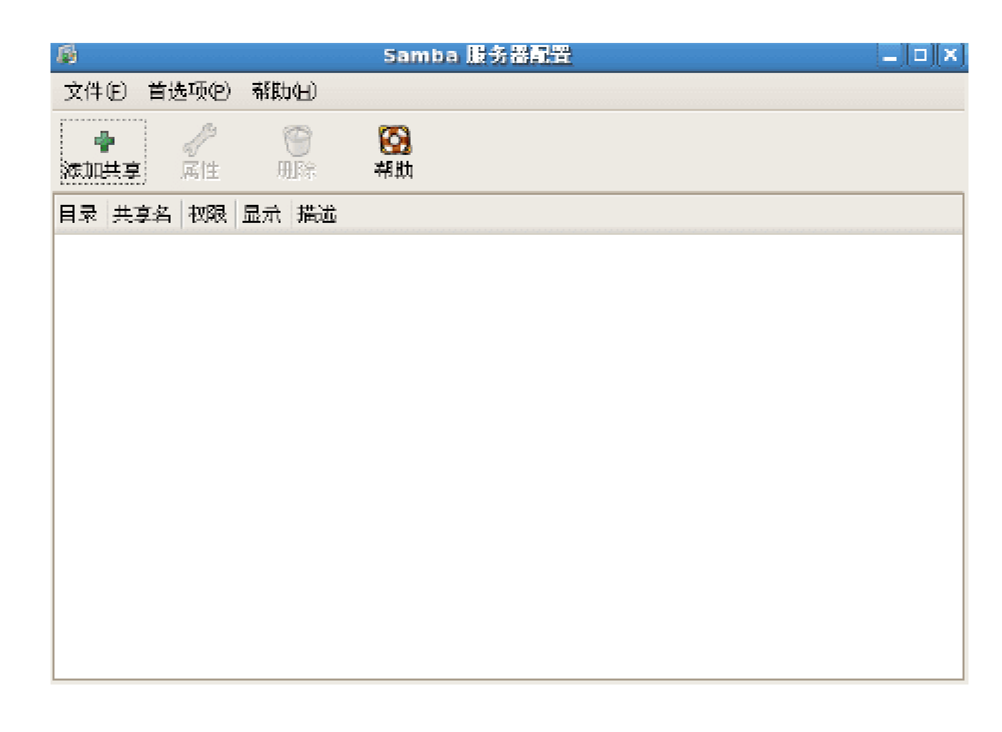


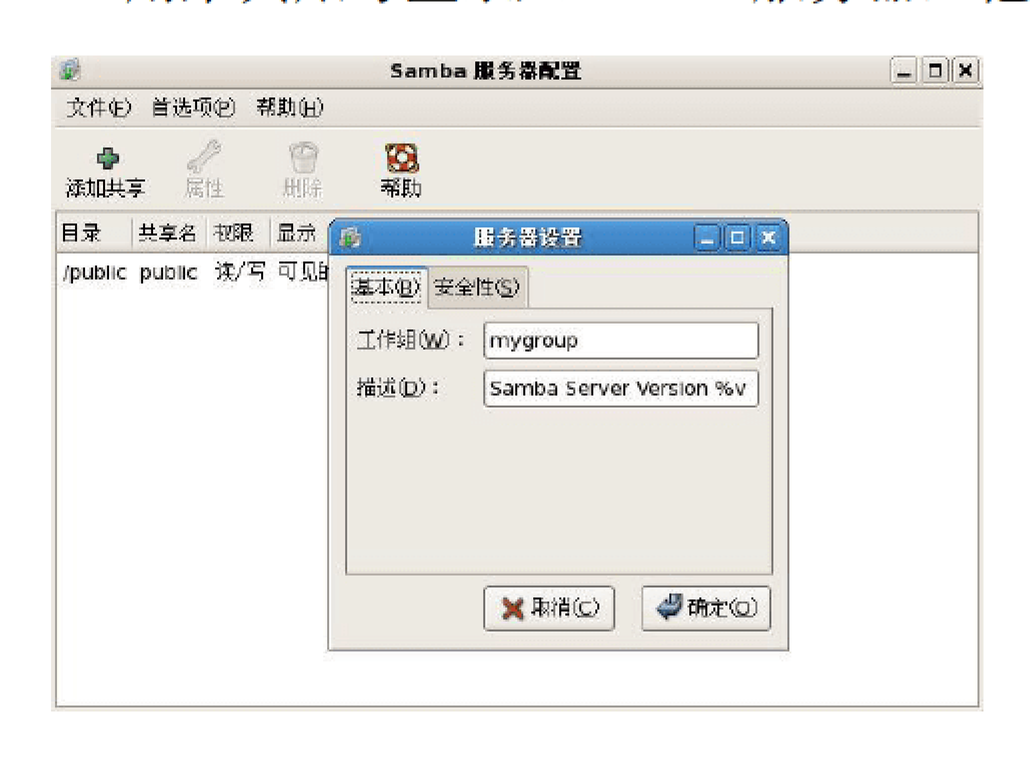
图 7-5　Samba 服务器配置界面

(3) 为方便他人识别这台计算机，需要先为这台计算机设置合适的描述和工作组。单击“首选项”→“服务器设置”命令，根据情况设置工作组名称及描述该 Samba 服务器的说明文字，如图 7-6 所示。

(4) 单击“安全性”选项卡，如图 7-7 所示。

- 验证模式：Samba 服务器所采用的验证方式其中各参数含义具体见 6.3.1 节的“security =”参数。

- 验证服务器：对于“用户”及“共享”验证模式，无须启用此项设置。
- Kerberos 域：只有启用 ADS 验证模式，才需要输入 Kerberos 域。对于“用户”及“共享”验证模式，无须启用此项设置。
- 加密口令：应该选择“是”，这样可以防止黑客用嗅探器截获密码明文。
- 来宾账号：当来宾用户要登录 Samba 服务器时，他们必须被映射到服务器上的某个有效用户。选择系统上的现存用户名之一作为来宾 Samba 账号。当用户使用来宾账号登录入 Samba 服务器，他们拥有和这个用户相同的权限。



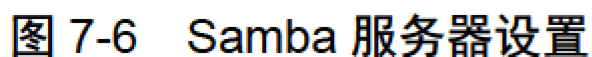

图 7-6 Samba 服务器设置



图 7-7 Samba 服务器安全性设置

(5) 添加共享目录。接下来添加共享目录，单击 Samba 配置窗口工具栏上的“添加共享”按钮，即可打开一个添加共享对话框如图 7-8 所示。在该对话框上的“基本”选项卡上，指定共享的目录为某个存在的目录，再指定共享名及描述，并设定该目录的基本权限是只读还是读/写。在“访问”选项卡上，可以指定允许所有用户访问，或者只允许某些用户访问。

(6) 添加 samba 共享用户。在“首选项”中选择“samba 用户”选项卡，启动 Samba 用户管理界面，如图 7-9 所示。单击“添加用户”可以实现将 Linux 系统中的已存在用户添加到 Samba 中来，并且需要为用户设置 Windows 用户名和 Samba 口令。选中某个用户单击“编辑用户”按钮可以对该用户的 Windows 用户名和 Samba 口令进行修改；选中某个用户单击“删除用户”按钮可将改用户自 Samba 服务器中删除。在 Samba 中删除用户对系统用户没有影响。但删除系统用户会将使 Samba 中所对应的用户在 Samba 重启启动后无法使用。



图 7-8 创建 Samba 共享



图 7-9 Samba 用户管理

(7) 启动 Samba 服务，在 Windows 或 Linux 客户端中对其进行验证测试。

注意： 必须是 root 用户才可以对 Samba 服务器进行配置。

7.8　Samba 客户端的配置

Samba 服务器配置完成后，Linux/Unix 操作系统的 Samba 客户可以在图像界面下通过在窗口地址栏输入“smb:\\服务器 IP\共享目录”访问 Samba 服务器的共享目录，在字符界面下可以通过 smbclient、smbmount 命令挂载 Samba 服务器目录并使用其中的资源。Windows 操作系统用户可以通过“网上邻居”或者直接在“我的电脑”地址栏输入“\\服务器 IP\共享目录”的方式访问 Samba 服务器资源。

7.8.1　Linux 客户端访问 Samba

Linux 客户端访问 Samba 服务器的方法很多，在图形界面下，访问 Samba 的方法如下。

(1) 在 Linux 图形界面的菜单栏选择“应用程序”→“系统工具”→“文件浏览器”即可打开文件浏览器，如图 7-10、图 7-11 所示。

(2) 单击按钮使其显示出地址栏，并在地址栏内输入“smb://192.168.0.103/alice”。如需要身份认证，则需要在弹出的对话框中输入用户名及密码，打开服务器资源如图 7-12 所示。



图 7-10　打开文件浏览器

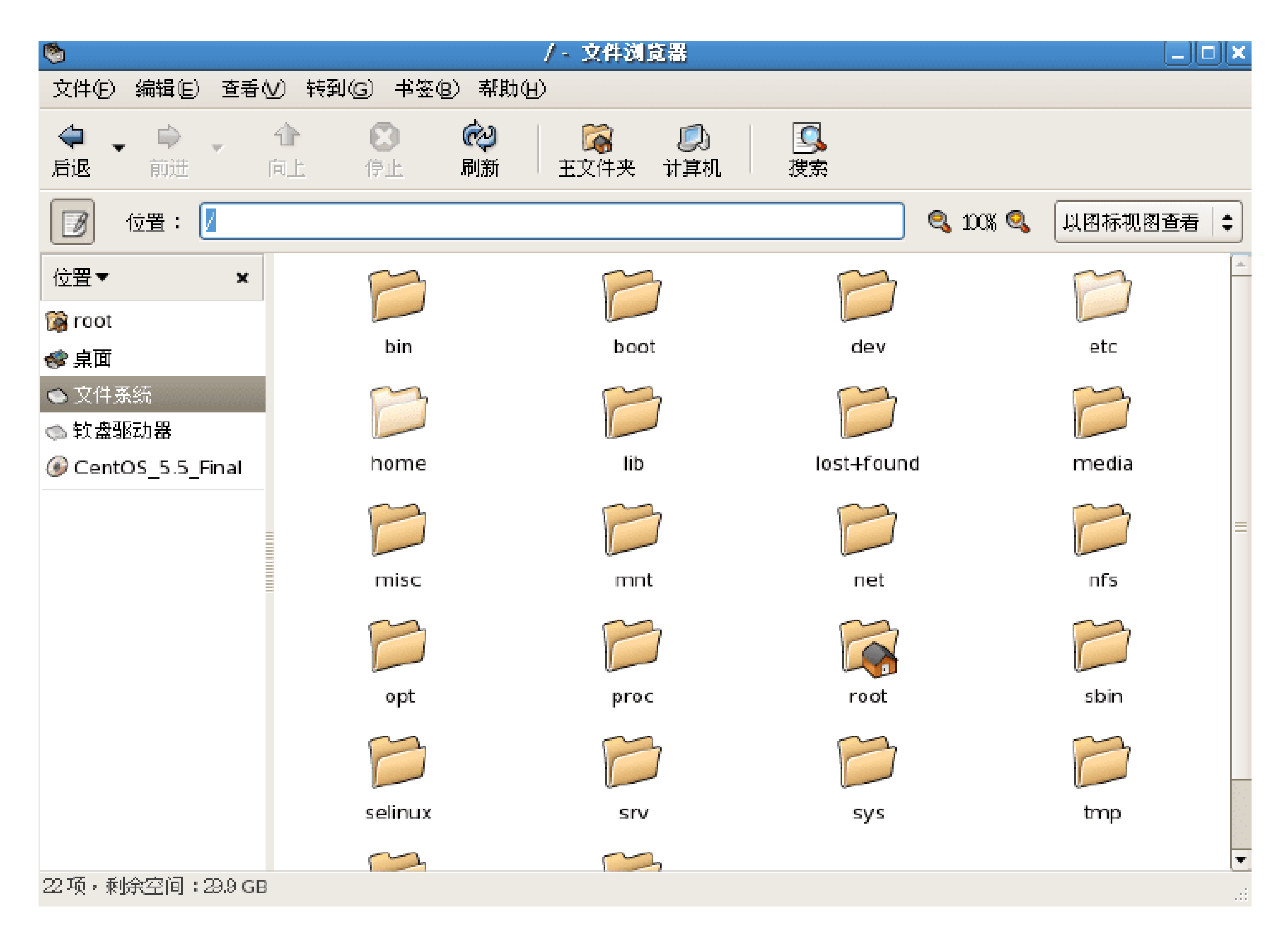


图 7-11　Linux 下的文件浏览器

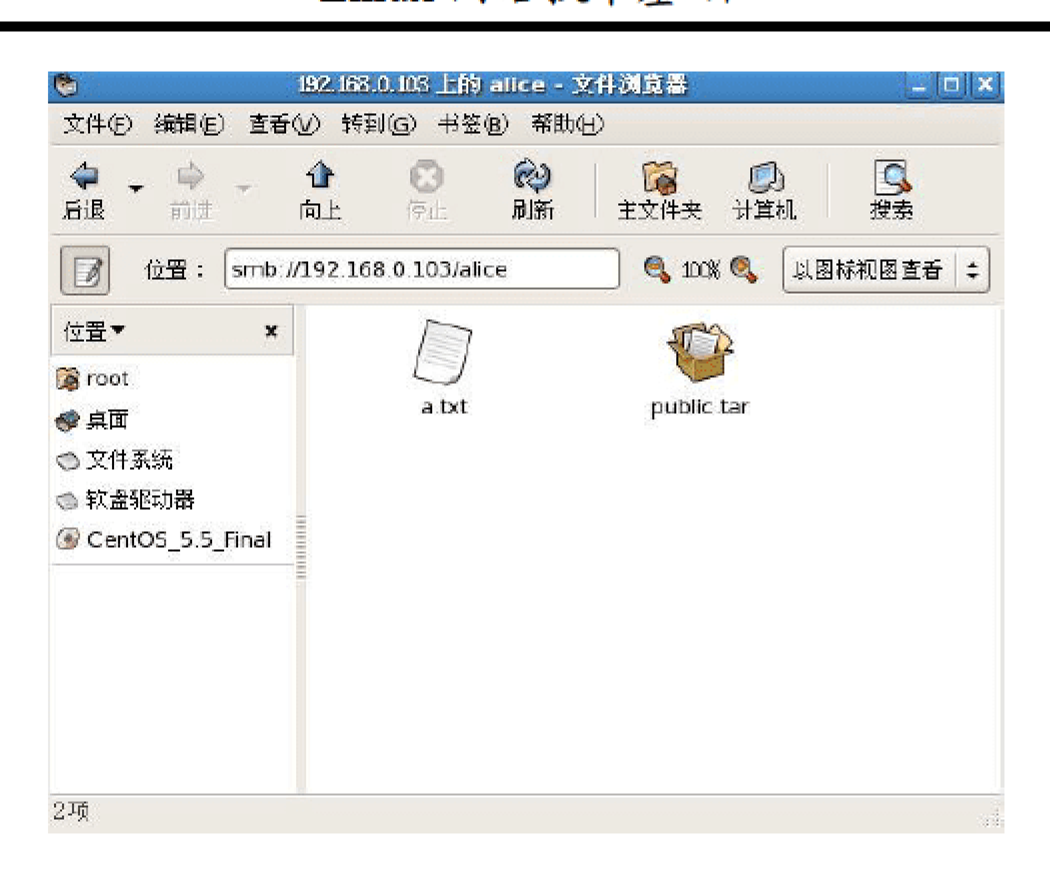


图 7-12　Linux 访问 Samba 资源

7.8.2　Windows 客户端访问 Samba

在 Windows 中访问 Samba 资源比较简单，直接在“我的电脑”地址栏输入“\\192.168.0.103\alice”按 Enter 键即可。在弹出的身份验证对话框中输入用户名与密码，即可访问到 Samba 服务器资源，如图 7-13 所示。

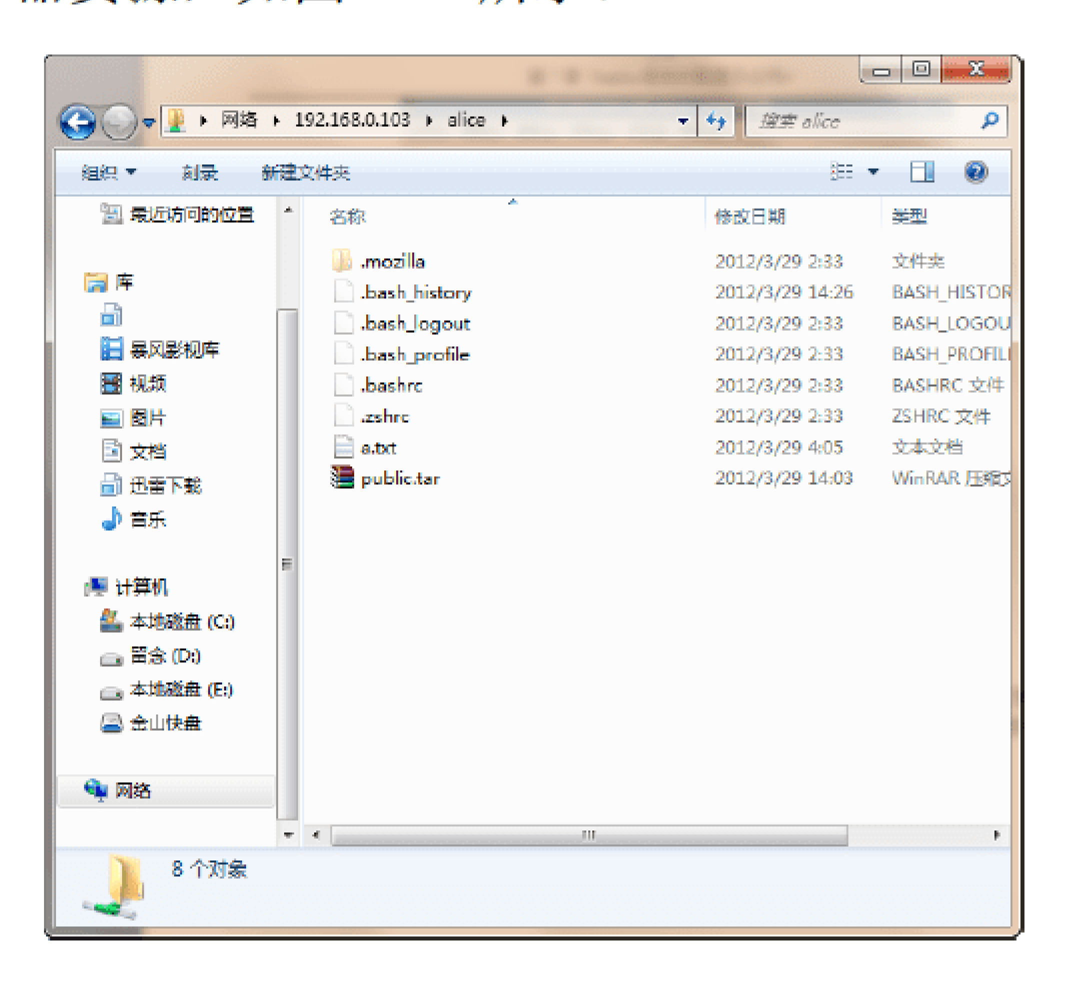


图 7-13　Windows 访问 Samba 资源

7.9　Samba 服务的配置案例

7.9.1　配置案例 1

【例 7-1】某一小型企包括经理共 8 名员工，所有员工在公司内流动办公。现公司要求员工不管在哪台电脑上网，都需要将文件保存在 Samba 服务器上。其中技术组和业务组各有一个自己的组共享目录，各组人员不得跨组访问。如 tom 和 jim 是业务组(sales)成员，lili 和 ann 是技术组(tech)成员，两组成员都只能访问各自的共享目录，不能相互访问。经理 bob 可以访问所有的共享目录。员工都可以访问自己的 home 目录。

实施步骤如下。

(1) 创建 5 个用户及两个工作组。Bob 作为经理同事隶属于两个组。首先向 Linux 系统中添加 sales 与 tech 用户组：

```
# groupadd sales
# groupadd tech
```

然后需要向系统中添加员工用户信息。

```
# useradd -g sales  tom
# useradd -g sales  jim
# useradd -g tech  lili
# useradd -g tech  ann
# useradd -G tech,sales bob
```

(2) 修改 Samba 主配置文件，添加共享目录。分别在根目录建立 sales 目录与 tech 目录，并更改目录权限，使用户能够正常对目录进行读写：

```
# mkdir /sales /tech
# chmod 777 /sales /tech

# vi /etc/samba/smb.conf
```

并在文件尾部添加内容如下：

```
[tech]
        comment = tech
        path = /tech
        valid users=@tech,bob
           #有效能够登录的用户为 tech 组用户与 bob
            write list=@tech,bob
           #有效能够登录的用户为 tech 组用户与 bob
            read only =No

[sales]
        comment = sales
        path = /sales
        valid users=@sales,bob
           #有效能够登录的用户为 sales 组用户与 bob
            write list=@sales,bob
           #有效能够登录的用户为 sales 组用户与 bob
            read only =No
```

实现每个员工都有自己的 home 目录，只要保持/etc/samba/smb.conf 中关于[home]目录选项不变即可：

```
[homes]
        comment = Home Directories
        browseable = no
        writable = yes
;       valid users = %S
;       valid users = MYDOMAIN\%S
```

(3) 创建 Samba 用户，将所有用户添加到 samba 服务器中：

```
# smbpasswd -a  jim
# smbpasswd -a  tom
# smbpasswd -a  lili
```

```
# smbpasswd -a  ann
# smbpasswd -a  bob
```

(4) 关闭防火墙、设置 SELinux 至 permissive 模式并启动 Samba 服务：

```
# service iptables stop
# setenforce 0
# service smb start
启动 SMB 服务：                                              [确定]
启动 NMB 服务：                                              [确定]
# ps -eaf  | grep smb
root      9796     1  0 16:54 ?        00:00:00 smbd -D
root      9799  9796  0 16:54 ?        00:00:00 smbd -D
root      9802  9796  0 16:55 ?        00:00:00 smbd -D
root      9815  9448  0 16:55 pts/3    00:00:00 grep smb
```

(5) 测试 bob 用户：

```
 # smbclient //192.168.0.103/bob -U bob%bob
Domain=[BOGON] OS=[Unix] Server=[Samba 3.0.33-3.28.el5]
smb: \>
# smbclient //192.168.0.103/sales -U bob%bob
Domain=[BOGON] OS=[Unix] Server=[Samba 3.0.33-3.28.el5]
smb: \>
# smbclient //192.168.0.103/tech -U bob%bob
Domain=[BOGON] OS=[Unix] Server=[Samba 3.0.33-3.28.el5]
smb: \>
```

由上面结果可知，经理 bob 用户对于个人目录、sales 目录、tech 目录均有访问权限。

(6) 测试 sales 组 tom 账号：

```
# smbclient //192.168.0.103/tom -U tom%tom
Domain=[BOGON] OS=[Unix] Server=[Samba 3.0.33-3.28.el5]
smb: \>
# smbclient //192.168.0.103/sales -U tom%tom
Domain=[BOGON] OS=[Unix] Server=[Samba 3.0.33-3.28.el5]
smb: \>
# smbclient //192.168.0.103/tech -U tom%tom
Domain=[BOGON] OS=[Unix] Server=[Samba 3.0.33-3.28.el5]
tree connect failed: NT_STATUS_ACCESS_DENIED
```

由上面结果可知，sales 组 tom 用户能访问个人目录、sales 目录，但是无法访问技术组共享目录 tech。

(7) 测试 tech 组 ann 账号：

```
# smbclient //192.168.0.103/ann -U ann%ann
Domain=[BOGON] OS=[Unix] Server=[Samba 3.0.33-3.28.el5]
smb: \>
# smbclient //192.168.0.103/tech -U ann%ann
Domain=[BOGON] OS=[Unix] Server=[Samba 3.0.33-3.28.el5]
smb: \>
# smbclient //192.168.0.103/sales -U ann%ann
Domain=[BOGON] OS=[Unix] Server=[Samba 3.0.33-3.28.el5]
tree connect failed: NT_STATUS_ACCESS_DENIED
```

由上面结果可知，tech 组 ann 用户能访问个人目录、tech 目录。但是无法访问业务组共享目录 sales。

7.9.2　配置案例 2

【例 7-2】某企业需要配置一台 CentOS 5 操作系统的文件服务器，使企业内员工可以实现文件资源共享及网络打印机共享。该企业的网络拓扑如图 7-14 所示，企业中客户端使用的 Windows XP 操作系统。其中设计部计算机使用 192.168.0.0/24 网段，市场部计算机使用 192.168.1.0/24 网段，财务部计算机使用 192.168.2.0/24 网段。该文件服务器 IP 地址为 192.168.0.2；FQDN 为 fs.example.com.，客户端已配置完成。每位员工用户建立完成并将/dev/sda10 挂载到/share 作为共享分区。企业对文件服务器的要求如下。

图 7-14　案例网络拓扑

- 财务部及设计部所有客户端，除公用计算机(pub.example.com)外，均可以使用该文件服务器。
- 设计部所有客户端可以使用文件服务器上的光驱。
- 需要一个存放内部资料的目录，所有用户只可读其中内容。
- 每个部门有一个需要一个存放资料的公用目录，只允许该部门员工可见/可读/可写。
- 每个部门的公用目录中的内容，除了上传文件的用户及管理员以外其他用户不能删除。
- 每位员工有一个可读写的目录，且该目录部门经理可读。

案例实施过程如下。

(1) 在 fs.example.com 上安装 Samba 服务。

```
#rpm -ivh samba-* system-config-samba-*
```

(2) 在 fs.example.com 上配置/dev/sda10 挂载参数，让该分区支持 acl 及硬盘配额。在/etc/fstab 中将/dev/sda10 的参数改为以下内容，修改完成后用 mount -o remount /share 命令重新载入挂载参数。

```
# /dev/sda10  /share  ext3      defaults,acl,usrquota,grpquota     0  0
```

(3) 在 fs.example.com 上根据部门建立用户组，并将用户加入相应组(//开头为注释语句)。

```
//设计部部门组
# groupadd design
//财务部部门组
# groupadd finance
//部门经理组
# groupadd manager
//以下是将设计部所有员工初始组设置为 design
# usermod -g design davidxu
# usermod -g design tomyang
以下是将财务部所有员工初始组设置为 finance
# usermod -g finance mikeliu
```

```
# usermod -g finance janeli
以下是将 manager 组加入财务部及设计部二位经理的额外组
# usermod -aG manager davidxu
# usermod -aG manager janeli
```

(4) 在 fs.example.com 上建立所需目录。

```
//内部资料公用目录。
# mkdir /share/public
//设计部专用目录。
# mkdir /share/design
//财务部专用目录。
# kdir /share/finance
//设计部公用目录。
# mkdir /share/design/public
//财务部公用目录。
# mkdir /share/finance/public

//以下是为每位员工建立专用目录
# mkdir /share/design/davidxu
# mkdir /share/design/tomyang
# mkdir /share/finance/mikeliu
# mkdir /share/finance/janeli

//以下是为通过 Sticky 实现每个部门公用目录除了上传文件的用户及管理员以外其他用户不#能删除
# chmod o+t /share/design/public
# chmod o+t /share/finance/public
```

(5) 在 fs.example.com 上将系统用户加入 Samba 服务器。

```
# smbpasswd -a davidxu
# smbpasswd -a tomyang
# smbpasswd -a mikeliu
# smbpasswd -a janeli
```

(6) 在 fs.example.com 上使用如下命令修改 SELinux 状态。

```
# setsebool -P samba_domain_controller on
# setsebool -P samba_enable_home_dirs on
# setsebool -P samba_export_all_rw on
# setsebool -P smbd_disable_trans=1
# chcon -R -t samba_share_t /share
# chcon -R -t samba_share_t /bin/mount
# chcon -R -t samba_share_t /bin/umount
```

(7) 在 fs.example.com 上修改 smb.conf，在[global]标签下加入如下内容。

```
    [global]
        hosts allow = lo 192.168.0. 192.168.2. EXCEPT 192.168.0.99
        include = /etc/samba/%G.smb.conf
        include = /etc/samba/%U.smb.conf
```

(8) 在 fs.example.com 上修改 smb.conf 文件，内容如下。

```
[public]
    path = /share/public

 [cdrom]
    path = /mnt/cdrom
    root preexec = /bin/mount -t iso9660 /dev/cdrom /mnt/cdrom
    root postexec = /bin/umount /mnt/cdrom
```

(9) 在 fs.example.com 上/etc/samba 目录下建立以下子配置文件。

修改 design.smb.conf 文件：

```
[design]
   path = /share/design/public
   write list = @design
```

修改 davidxu.smb.conf 文件：

```
[davidxu]
  path = /share/design/davidxu
    write list = davidxu

    [tomyang]
    path = /share/design/tomyang
   readonly = yes
```

修改 tomyang.smb.conf 文件：

```
[tomyang]
    path = /share/design/tomyang
   write list = tomyang
```

修改 finance.smb.conf 文件：

```
[finance]
    path = /share/finance/public
    write list = @finance
```

修改 mikeliu.smb.conf 文件：

```
[mikeliu]
    path = /share/finance/mikeliu
    write list = mikeliu

    [janeli]
    path = /share/finance/janeli
    read only = yes
```

修改 janeli.smb.conf 文件

```
[janeli]
    path = /share/finance/janeli
    write list = janeli
```

(10) 在 fs.example.com 上设置目录自身权限。

```
# setfacl -R -m d:g:design:rwx /share/design
# setfacl -R -m g:design:rwx/share/design
# setfacl -R -m d:g:finance:rwx /share/finance
# setfacl -R -m g:finance:rwx /share/finance
```

(11) 在 fs.example.com 上启动 Samba 服务，并设置为下次启动自动加载。

```
# service smb restart
# chkconfig smb on
```

到此 Samba 服务器的配置已可满足该企业的所有需求。

7.10 本 章 小 结

Samba 是 Linux/Unix 中最常用的文件共享软件之一。本章首先介绍了 Samba 概况、安装与配置文件；然后介绍了 Samba 服务器的具体配置参数、服务器启动停止方法和 Samba 下的常用工具命令；最后介绍了 Samba 服务器和客户端的具体配置方法。读者可根据两个配置案例进行试验，配置自己的 Samba 文件共享服务器。

7.11 课 后 习 题

1. 填空题

(1) Samba 服务的主配置文件路径及文件名是________。

(2) Linux 中图形界面配置 Samba 服务器的软件包是________。

(3) 使用 smbpasswd 命令向 Samba 服务器中添加用户的先决条件________。

(4) SMB 是________的缩写。

2. 选择题

(1) 某公司使用 Linux 系统搭建了 Samba 文件服务器，在账号为 benet 的员工出差期间，为了避免该账号被其他员工冒用，需要临时将其禁用，可以使用以下(　　)命令。

A. smbpasswd -a benet　　B. smbpasswd -d benet

C. smbpasswd -e benet　　D. smbpasswd -x benet

(2) 关于 Samba 用户账号，以下说法错误的是(　　)。

A. 使用 smbpasswd -a 添加的 Samba 账号必须已经是 Linux 的系统用户账号

B. 使用 smbpasswd -x 删除一个 Samba 用户时，同名的系统用户将会被锁定

C. Samba 用户和同名系统用户的口令可以不一致

D. 若 Samba 用户不需要登录 Linux 系统时，同名系统用户可以不设置口令

(3) Samba 服务器可以在 Linux/Unix 系统中提供 Windows 文件共享服务，在 RHEL4 系统中默认安装了 Samba 服务器和客户机所需要的软件包，在与 Samba 服务器相关的软件包中，(　　)是 Red Hat 公司专门为 Samba 服务器提供的配置工具。

A. samba-common　　B. samba　　C. samba-client　　D. system-config-samba

(4) 在 CentOS 5 中，用 samba 向 windows 提供共享服务时，使用用户认证来保证合法访问，下列关于 samba 用户描述正确的是(　　)。(选择三项)

A. samba 用户必须是系统用户　　B. 可以使用 smbuseradd 添加 samba 用户

C. samba 用户必须和系统用户同名　　D. 可以使用 smbpasswd 修改 samba 用户密码

(5) 你在某台 Windows 计算机上共享了一个文件夹，并想在 CentOS 5.5 计算机上访问它，可以使用的命令有(　　)。

A. mount　　B. smbclient　　C. share　　D. ftp

3. 简答题

(1) 简述 Samba 服务器配置步骤。

(2) 简述 Samba 服务器的优点。

第 8 章

WWW 服务的配置及应用

随着 Internet 上 Web 服务的发展，几乎各个政府部门、公司、大专院校、科研院所等都在构建或正在建设自己的网站，WWW(World Wide Web，万维网)已经成为 Internet 获取信息的主要途径。Web 服务是实现信息发布、资料查询、数据处理、视频点播等诸多应用的基本平台，所以架设 Web 服务器是 Internet 和 Intranet 必不可少的工作。本章将详细介绍 Web 的工作原理、HTTP 协议以及如何使用功能强大的 Apache 服务器软件来架设 Web 服务器。

8.1 WWW 服务概述

Web 服务是 Internet 中最为重要的应用，它是实现信息发布、资料查询、数据处理和视频点播等诸多应用的基本平台，并采用超级链接(Hypertext)的方式，将信息透过 Internet 传递到世界各处。Web 服务器作为网站的载体，是信息内容的发布者，最常见的客户端是浏览器，是信息内容的接收者。

8.1.1 HTTP 协议

网络的目的就是使信息更易于获取，而不管它们的地理位置在哪里。当使用超文本作为 WWW 文档的标准格式后，人们开发了可以快速获取这些超文本文档的协议——HTTP(HyperText Transfer Protocol)协议，即超文本传输协议。

HTTP 是应用层的协议，主要用于分布式、协作的信息系统。HTTP 协议是通用的、无状态的，其系统的建设和传输的数据无关。HTTP 也是面向对象的协议，可以用于各种任务，包括名字服务、分布式对象管理、请求方法的扩展、命令等。

在 Internet 上，HTTP 通信往往发生在 TCP/IP 连接上，其默认的端口为 80，也可以使用其他端口。

8.1.2 统一资源标识符 URI

Web 服务中的可用的每种资源，不论是 HTML 文档、图像、视频片段还是程序程序，都由一个通用资源标志符(Uniform Resource Identifier，简称“URI”)进行定位。

在 HTML 中，URI 主要有以下的作用：

- 链接到另一个文档或资源。
- 链接到一个外部样式表或脚本。
- 在页内包含图像、对象或 applet。
- 建立图像映射。
- 提交一个表单。
- 建立一个框架文档。
- 引用一个外部参考。
- 指向一个描述文档的 metadata。

URI 一般由以下三部分组成。

1. 存放资源的主机名

资源自身的名称，由路径表示。

例如下面的 URI，它使用当前的 HTML4.0 规范书写。

```
http://www.webmonkey.com.cn/html/html40/
```

这个 URI 的含义是：这是一个可通过 HTTP 协议访问的资源，位于主机 www.webmonkey.com.cn 中，通过路径“/html/html40”访问。在 HTML 文档中其他资源

包括“mailto”(收发 E-mail)和“ftp”(FTP 访问)。

又例如下面是一个表示用户 E-Mail 地址的 URI。

```
mailto:abc@163.com
```

注意： 很多读者可能熟悉 URL(Uniform Resource Locator，统一资源定位符)，而不是 URI。实际上，虽然两个词的名称相近，但表示的意思却有很大的差别，可以说，URL 只是 URI 命名机制的一个子集。

2. 片段标志符

有的 URI 指向一个资源的内部。 这种 URI 以“#”结束，并跟着一个 anchor 标志符(称为片段标志符)。例如，下面是一个指向 section_2 的 URI。

```
http://somesite.com/html/top.htm#section_2
```

3. 相对 URI

相对 URI 不包含任何命名规范信息。它的路径通常指同一台机器上的资源。相对 URI 可能含有相对路径(如“..”表示上一层路径)，还可能包含片段标志符。

为了说明相对 URI，假设我们有一个基本的 URI：http://www.acme.com/support/intro.htm。下面的链接中使用了相对 URI：

```
Suppliers
```

它扩展成完全的 URI 就是“http://www.acme.com/support/suppliers.htm”。

下面是一个图像的相对 URI：

```
<IMG src="../icons/logo.gif" alt="logo">
```

它扩展成完全的 URI 就是“http://www.acme.com/icons/logo.gif”。

8.1.3　Web 服务

Web 服务的实现采用客户/服务器(C/S)模型。客户机运行 WWW 客户程序——浏览器，它提供良好、统一的用户界面。浏览器的作用是解释和显示 Web 页面，响应用户的输入请求，并通过 HTTP 协议将用户请求传递给 Web 服务器。Web 服务器一端运行服务器程序，它最基本的功能是侦听和响应客户端的 HTTP 请求，向客户端发出请求处理结果信息。

Web 服务通常可以分为两种：静态 Web 服务和动态 Web 服务。

1. 静态 Web 服务

在静态 Web 服务中，服务器只是简单地负责把存储的文档发送给客户端浏览器，在此过程中传输的网页只有在网页编辑人员利用编辑工具对它们修改后，才会发生变化。

2. 动态 Web 服务

动态 Web 服务能够实现浏览器和服务器之间的数据交互。Web 服务器通过 CGI、

ASP、PHP 和 JSP 等动态网站技术，可以向浏览器发送动态变化的内容。在此过程中，服务器根据客户端浏览器发出的不同请求，在服务器端执行程序，组织好文档后再将结果发送至客户端。

8.2 HTTP 服务的工作原理

作为 Web 系统最核心的内容，HTTP 协议是一个在 TCP/IP 协议基础上的应用程序级协议。它基于请求/响应范式。一个客户端与服务器建立连接后，发送一个请求给服务器，请求消息的格式包括 URI、协议版本号和 MIME 信息，MIME 信息包括请求修饰符、客户机信息和其他内容。服务器接收到请求后，将给予相应的响应信息，其格式包括 HTTP 的协议版本号、一个成功或错误代码以及 MIME 信息，MIME 信息包括服务器信息、实体信息和可能的内容。

8.2.1 HTTP 的通信过程

最简单的 HTTP 通信方式是由用户代理和源服务器之间通过一个单独的连接来完成，如图 8-1 所示。

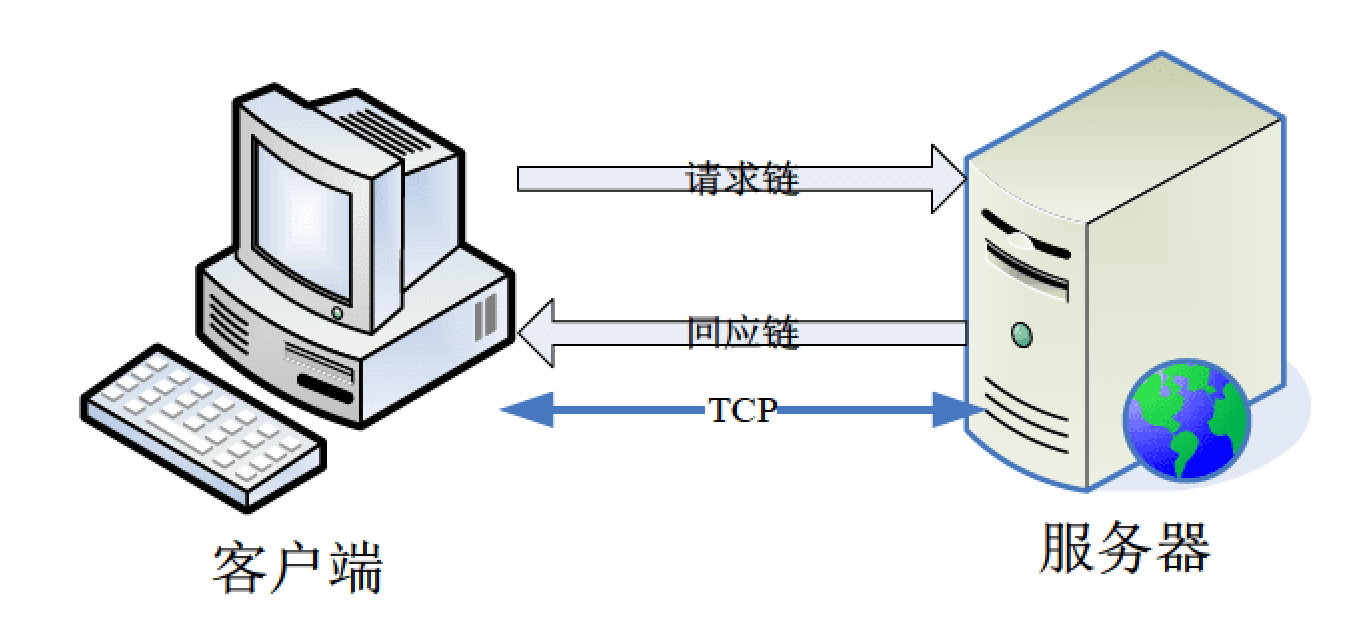


图 8-1　简单的 HTTP 连接

客户端的用户代理首先向服务器发起连接请求，源服务器接收请求后建立 TCP 连接，然后客户端通过建立的 TCP；连接提交一个申请服务器资源的请求链，如果服务器能够满足请求链的要求，就回应给客户端一个响应链。

但往往服务器和客户端之间并不能直接连接，中间会出现一个或者多个中介，这时，情况就变得复杂一些，如图 8-2 所示。

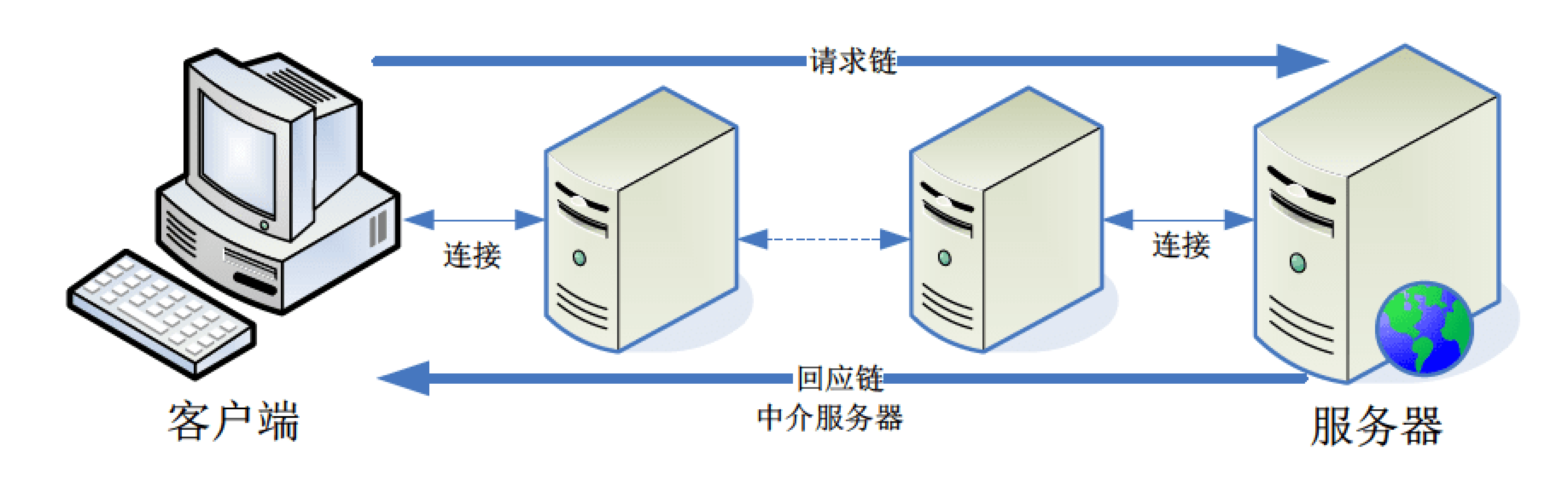


图 8-2　多个中介的 HTTP 连接

此时，客户端和服务器之间的数据通道不是直接连通的，而是通过中介进行转发。也就是说，客户端的请求链要经过中介才能到达服务器，而服务器的回应链也需要经过中介才能到达客户端。虽然图中的连接是线性的，但是每一个节点都可能从事多重并发的通信，即同时处理多个请求链或者回应链。

在 HTTP 协议中，中介包括以下 3 种类型。

1. 代理(Proxy)

代理根据 URI 的绝对格式来接受请求，重写全部或者部分消息，通过 URI 标识将已修改的请求发送到服务器。

2. 网关(Gateway)

网关是一个接收代理，作为一些其他服务器的上层，并且如果需要，可以将请求翻译给下层的服务器协议。

3. 通道(Tunel)

通道不会改变消息内容，只是两个连接之间的中继点，当通信只需要简单的穿过中介或者中介不需要识别消息内容时，经常使用通道作为中介。

另外，除了通道外，代理和网关还具有内部缓存的功能，即代理或者网关可以将回应链中的资源暂时存储在本服务器的缓存中，如果有客户端再次请求该资源，那么该中介不需要再将请求链发送到所请求的服务器中，而是直接可以将之前缓存的资源发送给客户端。这样不但响应速度加快，也节省了大量的网络资源。当然，缓存的数据还需要及时更新，才能与源服务器数据保持一致。

8.2.2 HTTP 的请求行和应答行

在 HTTP 协议中，一个通信信息的交换过程主要分为 5 个阶段：

(1) Web 浏览器使用 HTTP 命令向一个特定的服务器发出 Web 页面请求消息。

(2) 若该服务器在特定端口(通常是 TCP 80 端口)处接收到 Web 页面请求后，就发送一个应答并在客户和服务器之间建立连接。

(3) 服务器 Web 查找客户端所需文档，若 Web 服务器查找到所请求的文档，就会将所请求的文档传送给 Web 浏览器。若该文档不存在，则服务器会发送一个相应的错误提示文档给客户端。

(4) Web 浏览器接收到文档后，就将它显示出来。

(5) 当客户端浏览完成后，就断开与服务器的连接。

HTTP 协议的请求消息的格式如下：

```
<请求行>
[通用头域]
[请求头域]
[实体头域]
CR/LF
[实体数据]
```

HTTP 协议的应答消息的格式如下：

```
<应答行>
[通用头域]
[应答头域]
[实体头域]
CR/LF
[实体数据]
```

在以上的格式中，每一种头域都可以有一个或者多个成员，以“域名：域值”的形式给出。请求行由 3 部分组成：请求方法、URI 和 HTTP 版本，之间以空格分隔，例如下面的代码是一个典型的请求行：

```
GET http://httpd.apache.org/docs/2.2/license.html HTTP/1.1
```

其中，GET 是请求方法，http://httpd.apache.org/docs/2.2/license.html 是 URI，HTTP/1.1 是协议版本。在 HTTP 规范中一共定义了 8 种可能的请求方法，其名称和含义如表 8-1 所示。

表 8-1 HTTP 协议请求方法及其含义

请求方法	含 义
GET	检索 URI 所标识的资源
HEAD	与 GET 方法相同，但只要求返回状态行和头域，并不返回所请求的内容
POST	向服务器发送数据，请求服务器接收
PUT	服务器保存请求数据作为制定 URI 新内容的请求
DELETE	请求服务器删除指定 URI 中命名的资源
OPTIONS	请求得到服务器所支持的请求方法
TRACE	用于调用已请求消息的远程、应用层回送信息
CONNECT	预留所谓隧道功能，还未实现

在这 8 种请求方法中，GET、HEAD、POST 方法是大部分 Web 服务器都支持的。其他的几种方法处于安全等原因的考虑，很少得到支持。

应答消息的应答行也由 3 部分组成：HTTP 版本、响应代码和响应描述，他们之间使用空格隔开，一个典型的应答行实例如下面的代码所示：

```
HTTP/1.1 200 OK
```

其中，HTTP/1.1 代表服务器可以接受的最高协议版本。200 为响应代码，所有的响应代码由 3 位数字组成，指出请求的成功或者失败，如果失败则会给出原因。响应代码的规定及具体含义如表 8-2 所示。

表 8-2 HTTP 协议响应代码及其含义

响应代码	含 义
1XX	信息，请求收到，继续处理
2XX	成功，行为被成功的接受、理解和采纳

续表

响应代码	含　义
3XX	重定向，为了完成请求，必须进一步执行的动作
4XX	客户端错误，请求包含语法错误或者请求无法实现
5XX	服务器端错误，服务器不能实现此请求

8.2.3　持久连接和非持久连接

在客户端向服务器发出请求之前，首先必须与服务器建立 TCP 连接。在 HTTP 协议中规定，TCP 连接既可以是非持久的连接，也可以是持久的连接，而具体采用哪种方式是由通用头域中的 Connection 值来决定的。在 HTTP/1.0 版本中，默认使用的是非持久连接，而 HTTP/1.1 版本默认使用的是持久连接。

1. 非持久连接

下面我们举一个例子来说明非持久连接是如何工作的。假设客户端以非持久连接向服务器请求传送一个 Web 页面，这个页面由 1 个 HTML 文件和若干个图片文件组成，这些文件都存放在同一台服务器中，假设该网页 URL 地址为 http://www.163.com/index.html。那么，在非持久连接的传输过程如下：

(1) 客户端初始化一个与主机 www.163.com 中的 Web 服务器的 TCP 连接，服务器使用默认端口号 80 接受此连接请求。

(2) HTTP 客户端经由与 TCP 连接相关的本地套接字发出一个 HTTP 请求，这个请求中包含路径名/index.html。

(3) Web 服务器接受此套接字请求消息，从服务器的主机内存或者硬盘中取出请求对象/index.html，经由同一个套接字发出包含该对象的应答消息。

(4) Web 服务器告知本地的 TCP 协议栈关闭这个 TCP 连接。

(5) HTTP 客户端接收应答消息，并同意断开 TCP 连接。

(6) 客户端从应答消息中的头域内容取出这个 HTML 文件，从中分析后发现还需要若干个图片文件才能正常显示该网页。

(7) 客户端重复(1)～(5)的步骤，从服务器中得到所需要的每一个图片文件。这时，整个请求网页的过程完成。

从上述的步骤可以看出，非持续连接在每次服务器发送一个对象后，相应的 TCP 连接就被关闭，也就是说每一个连接都没有持续到可用于传送其他的对象。每一个 TCP 连接只用于传送一个请求消息和一个应答消息。

2. 持久连接

于非持久连接不同，持久连接在服务器发出每一个响应后都可以继续保持 TCP 连接，同一对客户端和服务器的后续请求和响应都可以通过这个连接继续发送。不仅这个 Web 页面可以通过一个持久连接发送，而且还可以将服务器中的多个 Web 页面都通过这个 TCP 连接发送。

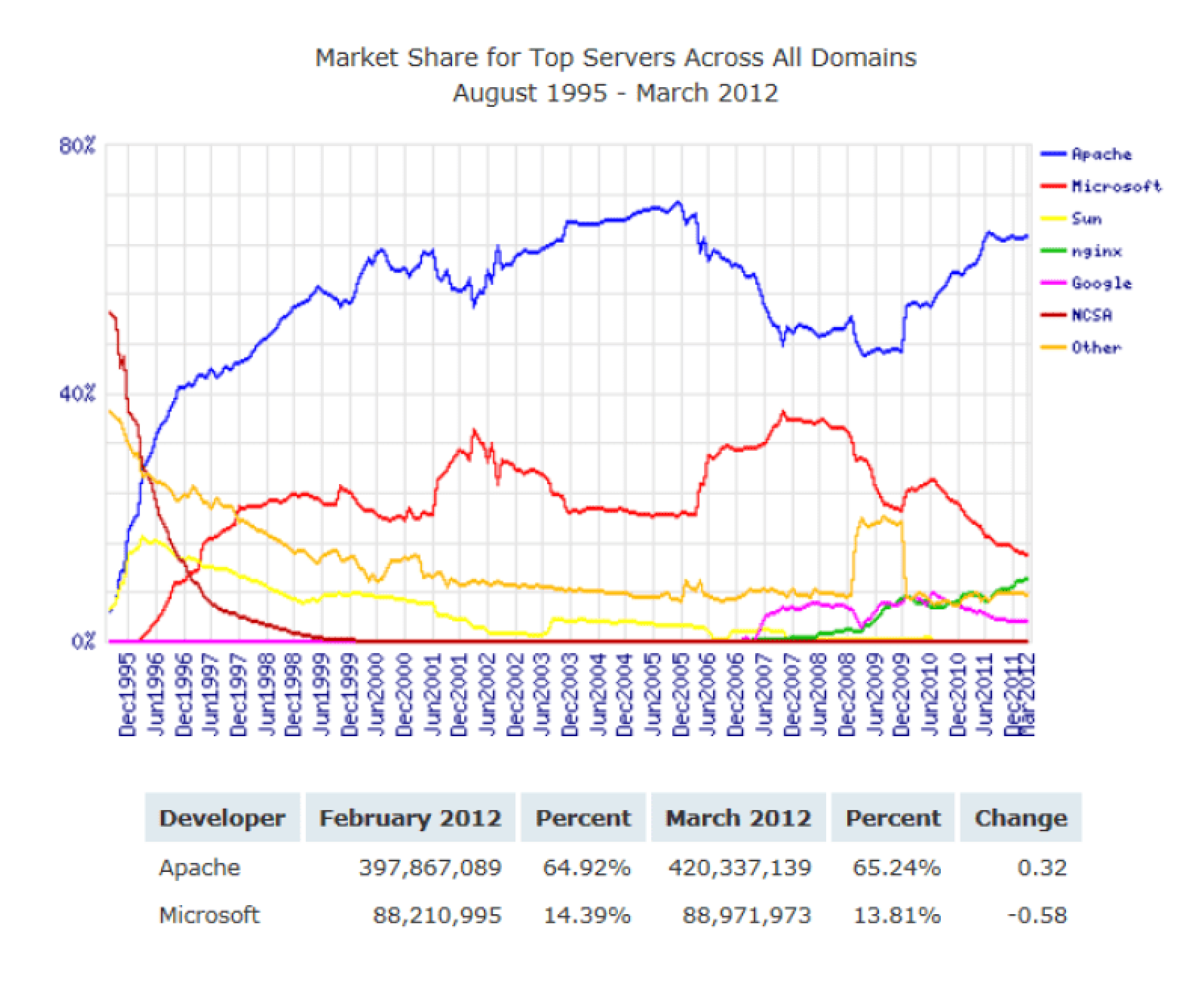
3. 非持久连接与持久连接的比较

通过对比，我们可以看出，非持久连接有些缺点。首先，客户端需要为每个请求对象建立并维护一个新的连接，这需要占用客户端和服务器的资源。对于可能同时为成千上万个不同客户端提供服务的 Web 服务器来说。这会严重增加负担。另外，建立 TCP 连接是需要耗费一定的时间，这也使访问 Web 服务器的速度减慢。而持久连接就很好地改善了以上的两个问题。

8.3　Apache 简介

开放源代码的 Apache(阿帕奇)服务器起初由 Illinois 大学 Urbana-Champaign 的国家高级计算程序中心开发，后来 Apache 被开放源代码团体的成员不断地发展和加强。开始时，Apache 只是 Netscape 网页服务器(现在是 Sun ONE)的之外的开放源代码选择。渐渐地，它开始在功能和速度上超越其他 Web 服务器。由于 Apache 服务器拥有牢靠可信的美誉，因此从 1995 年 1 月以来，Apache 一直是 Internet 上最流行的 Web 服务器。

根据著名的 Web 服务器调查公司 Netcrafe(http://www.netcraft.com)的最新调查，截止到 2012 年 3 月，Apache 的市场占有率为 65.24%，而同期 IIS 的市场占有率只有 13.81%，如图 8-3 所示。不难看出 Apache 依然是目前使用最广泛的 Web 服务器，并且市场占有率在还在不断的攀升。



Developer	February 2012	Percent	March 2012	Percent	Change
Apache	397,867,089	64.92%	420,337,139	65.24%	0.32
Microsoft	88,210,995	14.39%	88,971,973	13.81%	-0.58

图 8-3　Web 服务器类型市场占有率

为什么 Apache 能保持如此高增长速度并且得到如此广泛的应用呢？这与 Apache 自身的优点是分不开的。其主要优点有以下几个方面：

- 支持 HTTP/1.1 协议：Apache 是最先使用 HTTP/1.1 协议的 Web 服务器之一，它

完全实现了 HTTP/1.1 协议并与 HTTP/1.0 兼容。

- 支持通用网关接口(CGI)：Apache 使用 mod_cgi 模块来支持 CGI 功能。在遵守 CGI/1.1 标准的同时还提供了扩充的特征如定制环境变量功能以及很难在其他 Web 服务器中找到的调式支持功能。
- 支持 HTTP 认证：Apache 支持基于 Web 的基本认证，它还为支持基于消息摘要的认证做好了准备，Apache 可以使用标准的密码文件，也可以通过对外部认证程序的调用来实现基本的认证功能。
- 集成 Perl 语言：Perl 已成为 CGI 脚本编程的基本标准，这与 Apache 的支持是分不开的。通过 mod_perl 模块的调用，Apache 可以将基于 Perl 的 CGI 脚本装入内存，并可以根据需要多次重复使用该脚本，从而消除了执行解释性语言时的启动开销。
- 集成的代理服务器：Apache 可以作为前向代理服务器，也可以作为后向代理服务器。
- Apache 在监视服务器本身状态和记录日志方面提供了很大的灵活性，可以通过 Web 浏览器来监视服务器的状态，也可以根据自己的需要来定制日志。
- Apache 支持虚拟主机：即通过在一台服务器上使用不同的主机名来提供多个 HTTP 服务。Apache 支持包括基于 IP、名称、和 Port 三种类型的虚拟主机服务。
- Apache 的模块可以在运行时按需动态加载，避免了不需要的程序代码占用内存空间。
- Apache 支持安全套接字(SSL)。
- 用户对话过程的跟踪功能：通过使用 HTTP Cookies， mod_usertrack 模块可以在用户浏览 Apache 的 Web 站点时对其进行跟踪。
- 支持 Java Servlets：Apache 的 mod_jserv 模块支持 Java Servlets，这项功能可以使 Apache 服务器支持 Java 应用程序。
- 支持多进程：当负载增加时，服务器会快速生成子进程来应对，从而提供系统的响应能力。

提示： 虽然官方网站对 Apache 名字的解释是“Apache 这个名字是为了纪念名为 Apache(印地语)的美洲印第安人土著的一支，众所周知他们拥有高超的作战策略和无穷的耐性”，可是流传最广的解释是(也是最显而易见的)，这个名字来自这么一个事实，当 Apache 在 1995 年初开发的时候，它是由当时最流行的 HTTP 服务器 NCSA HTTPd 1.3 的代码修改而成的，因此是“一个修补的(a patchy)”服务器。

8.4　Apache 服务器的安装及运行

目前几乎所有的 Linux 发行版都捆绑了 Apache，CentOS 也不例外，但默认情况下安装程序不会将 Apache 安装在系统上。由于目前 Apache 被重新命名为 httpd，因此可使用

下面的命令检查系统是否已经安装了 Apache 或查看已经安装了何种版本。

```
[root@CentOS ~]# rpm -q httpd
httpd-2.2.3-43.el5.centos
```

从显示内容可以看出，CentOS 5.5 默认已经安装的 Apache 版本为 2.2.3。如果 Linux 没有默认安装，可以使用 CentOS 安装光盘中的 RPM 安装包来安装 Apache，使用以下的命令：

```
[root@CentOS ~]# rpm -ivh httpd-2.2.3-43.el5.centos.i386.rpm
```

安装成功后，生成的主要文件及目录有：

- /etc/httpd/conf/httpd.conf：Apache 的主配置文件。
- /etc/httpd/logs：Apache 日志的存放目录。
- /etc/httpd/modules：Apache 模块存放目录。
- /usr/lib/httpd/modules：Apache 模块也会存放在此目录。
- /usr/sbin/apachectl：Apache 控制脚本，用于 Apache 的启动、停止、重启等操作。
- /usr/sbin/httpd：Apache 服务器的进程程序文件。
- /usr/shar/doc/httpd-2.2.3：Apache 的说明文档存放位置。
- /var/www：Apache 提供的例子网站。

另外，为了便于学习，我们还可以安装 Apache 的帮助手册包，名为 httpd-manual-2.2.3-43.el5.centos.i386.rpm，可以使用以下命令来安装：

```
[root@CentOS ~]# rpm -ivh httpd-manual-2.2.3-43.el5.centos.i386.rpm
```

安装完成后，在/var/www/manual 目录下会添加网页形式的帮助手册，这些帮助手册可以与例子网站很好的结合，帮助读者学习 Apache 的使用。

RPM 包安装完成后，Apache 使用默认的配置即可运行，使用以下的命令可以启动 httpd 进程。

```
[root@CentOS usr]# service httpd start
Starting httpd: httpd:                                     [  OK  ]
```

启动后，我们首先来查看以下 Apache 都启动了哪些进程，使用以下命令来查看 httpd 进程。

```
[root@CentOS ~]# ps -eaf | grep httpd
root      4724     1  2 15:22 ?       00:00:00 /usr/sbin/httpd
apache     4727  4724  0 15:22 ?       00:00:00 /usr/sbin/httpd
apache     4728  4724  0 15:22 ?       00:00:00 /usr/sbin/httpd
apache     4729  4724  0 15:22 ?       00:00:00 /usr/sbin/httpd
apache     4730  4724  0 15:22 ?       00:00:00 /usr/sbin/httpd
apache     4731  4724  0 15:22 ?       00:00:00 /usr/sbin/httpd
apache     4732  4724  0 15:22 ?       00:00:00 /usr/sbin/httpd
apache     4733  4724  0 15:22 ?       00:00:00 /usr/sbin/httpd
apache     4734  4724  0 15:22 ?       00:00:00 /usr/sbin/httpd
root      4738  4606  0 15:22 pts/0    00:00:00 grep httpd
```

可以看到，在初始状态系统一共启动了 9 个 httpd 进程，其中一个是以 root 身份启动的，其他进程则是以 apache 用户的身份运行，而且以是 root 身份运行的进程的子进程。

提示： 初始进程数可以在配置文件中设置。

下面我们再查看一下 Apache 是否已经开始监听 TCP 的 80 端口，使用以下命令。

```
[root@CentOS usr]# netstat -an | grep :80
tcp        0      0 :::80                       :::*                        LISTEN
```

可以看到，80 端口已经处于监听状态。另外，为了保证客户端能够访问 Apache 服务器，还需要在防火墙中开放对 80 端口的限制，使用以下的命令。

```
[root@CentOS usr]# iptables -I INPUT -p tcp --dport 80 -j ACCEPT
```

执行命令后，确认 Web 服务已经启动后，在客户端使用的 Web 浏览器中输入 Linux 服务器的 IP 地址进行访问，如果出现 Apache 的测试页面，则表示 Web 服务安装正确并且运行正常，如图 8-4 所示。

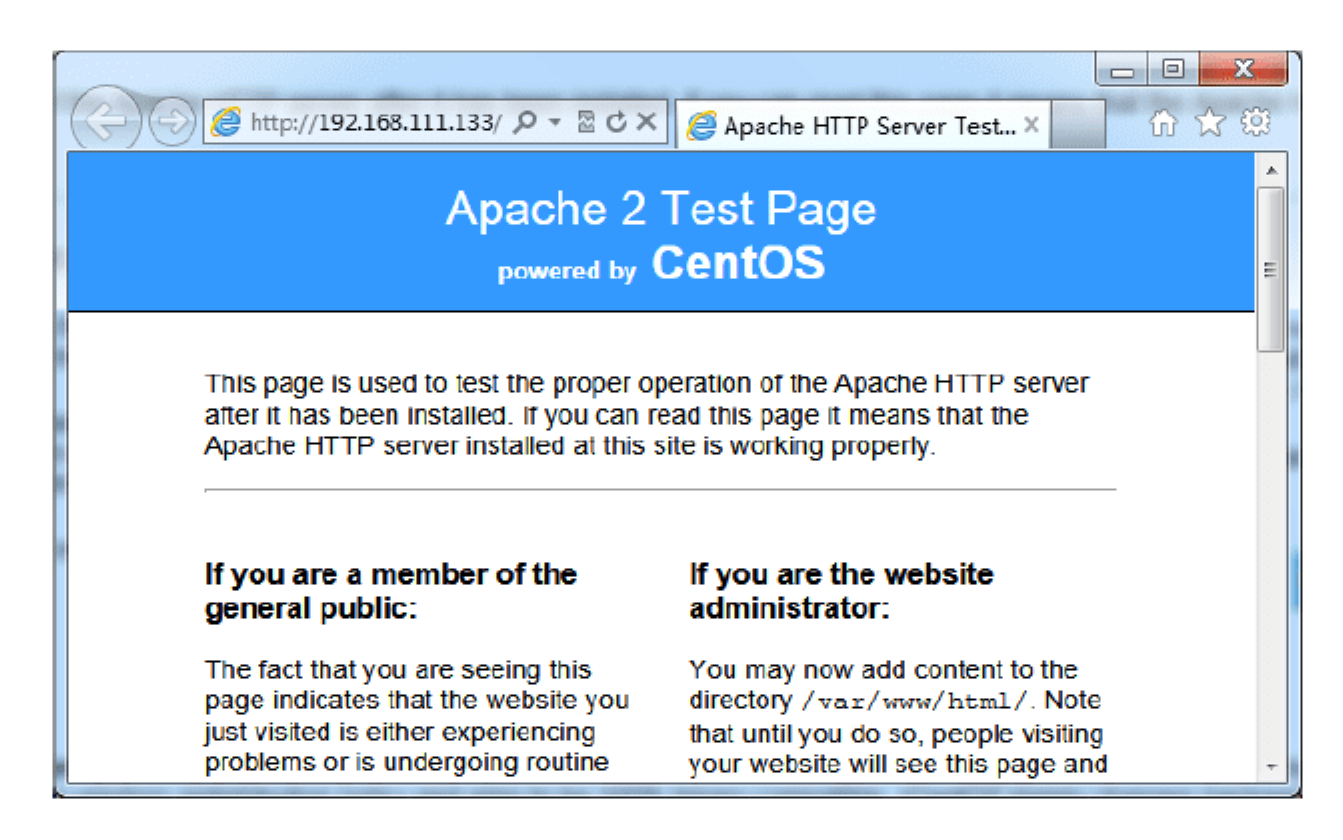


图 8-4　Apache 测试页面

另外，我们也可以访问主页的/manual 目录来查看 Apache 手册中的内容，如图 8-5 所示，其中包含了所有 Apache 配置命令的详细解释。

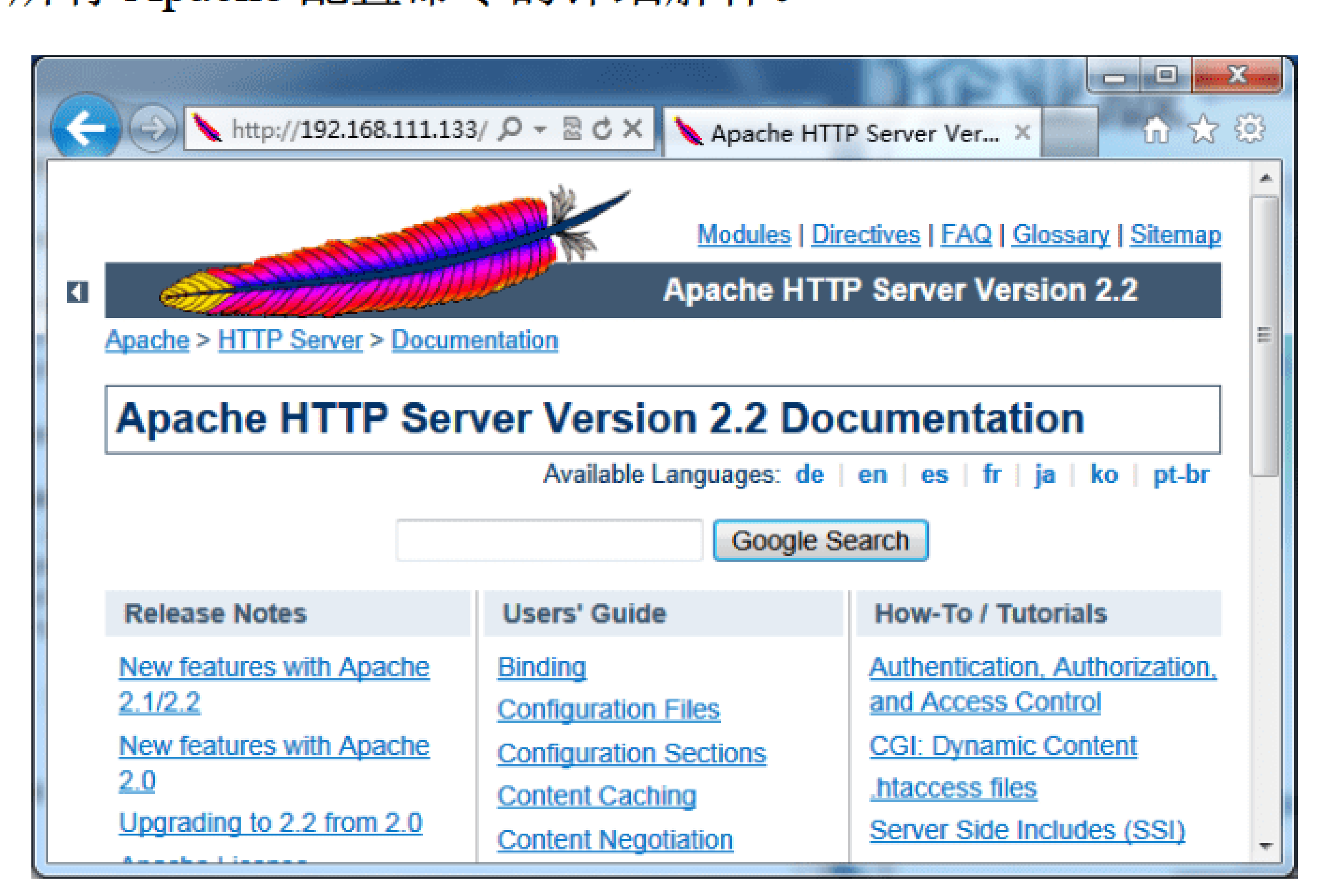


图 8-5　Apache 手册主页

至此，Apache 服务器的安装已经完成，不过处于初始状态的 Apache 服务器并不能很好的应用于 Web 服务，需要进行进一步的配置，我们将在下面的内容中详细介绍 Apache

服务器的配置方法。

8.5　Apache 服务器的基本配置

配置 Apache 服务器的运行参数，是通过编辑 Apache 的主配置文件 httpd.conf 来实现的。该文件的位置随着安装方式的不同而不同，如果使用 RPM 的方式安装，该文件通常存放在/etc/httpd/conf 目录下；如果使用编译源代码的方式安装，该文件通常存放在 Apache 安装目录的 conf 子目录下。由于 httpd.conf 是一个文本文件，因此可以使用任何文本编辑器(如 vi)对其进行编辑。

httpd.conf 配置文件主要由全局环境(Global Environment)、主服务器配置(“Main” Server Configuration)和虚拟主机(Virtual Hosts)3 个部分组成。每部分都有相应的配置语句，该文件所有配置语句的语法为“配置参数名称 参数值”的形式，配置语句可以放在文件中的任何地方，但为了增强文件的可读性，最好将配置语句放在相应的部分。

8.5.1　全局环境配置

下面我们先来解释一下配置文件中有关全局的配置参数。这些参数决定了 Apache 服务器的总体性能。

```
### Section 1: Global Environment
#当服务器响应主机头(header)信息时显示 Apache 的版本和操作系统名称
ServerTokens OS
#设置服务器的根目录
ServerRoot "/etc/httpd"
设置运行 httpd 进程时使用的 Pid 文件的路径和名称
PidFile run/httpd.pid
设置 TCP 连接的超时时间，如果 TCP 连接在此时间内没有收到或者发送任何数据则断开连接
Timeout 120
#设置是否使用持久连接，默认值为“Off”(关闭)，建议用户将此选项设置为“On”，这样就开
启了持久连接，以提高服务器的性能
KeepAlive Off
#在使用持久连接时，设置客户端通过该连接发送的最大请求消息数，如果设置为 0，表示没有限制
MaxKeepAliveRequests 100
#在使用持久连接时，设置客户端下一个请求消息超过 15 秒还未到达，就断开连接
KeepAliveTimeout 15

#设置使用 preforkMPM 运行方式的参数，此运行方式为默认方式，它规定 Apache 运行时，启动
多少个 httpd 子进程来处理客户端的请求
<IfModule prefork.c>                #启动 prefork 模块
StartServers        8               #设置起始 httpd 子进程总数
MinSpareServers      5              #最小空闲 httpd 子进程总数
MaxSpareServers     20              #最大空闲 httpd 子进程总数
ServerLimit       256               #最大 httpd 子进程允许总数
MaxClients        256               #最大客户端连接数
MaxRequestsPerChild  4000           #每一个 httpd 子进程处理了 4000 个请求后要关闭
</IfModule>                         #模块定义结束
#设置 worker 参数。但设置的是线程数
<IfModule worker.c>
```

```
StartServers        2              #主控制进程生成 httpd 子进程数
MaxClients        150              #最大客户端连接数
MinSpareThreads    25              #最小空闲线程总数
MaxSpareThreads    75              #最大空闲线程总数
ThreadsPerChild    25              #每一个子进程可产生的线程数
MaxRequestsPerChild  0             #每一个子进程可处理的最大请求数，0 表示无限制
</IfModule>

#设置服务器的监听端口可以使用“IP+端口”的形式规定监听哪一个本地接口的端口
Listen 80
#加载的动态模块(DSO)，默认加载了很多模块，在此不一一列出
LoadModule auth_basic_module modules/mod_auth_basic.so
LoadModule auth_digest_module modules/mod_auth_digest.so
LoadModule authn_file_module modules/mod_authn_file.so
……

#读取 conf.d 目录中的所有 conf 文件，这个目录中包含了许多专用功能的配置，如 PHP、SSL
等的配置文件
Include conf.d/*.conf
#此选项需要在 mod_status 模块加载时才有效，表示服务器是否为每一个请求保持扩展状态的轨迹
#ExtendedStatus On
#指定运行 httpd 子进程的用户和用户组
User apache
Group apache
```

提示： 在默认的 httpd.conf 文件中，每个配置语句和参数都有详细的英文解释，由于篇幅所限，此处省略了这些解释。建议初学者在不熟悉配置方法的情况下，先使用 Apache 默认的 httpd.conf 文件作为模板进行修改设置，并且在修改之前先做好备份，以便做了错误的修改后能够还原。

8.5.2　主服务器配置

Apache 在处理客户端的请求时，会根据 URL 来判断客户端是在访问主服务器还是在访问虚拟主机。所谓主服务器一般是指在一个服务器中提供的唯一的 Web 服务。下面是 Apache 配置文件中有关主服务器的配置命令，这些选项决定了主服务器的工作状态，同时，也可以作为虚拟主机的默认配置，如果在虚拟主机中没有设置这些参数，那么虚拟主机会读取主服务器的参数作为虚拟主机的参数。

```
### Section 2: 'Main' server configuration
#管理员的联系方式，会出现在一些错误页面中
ServerAdmin root@localhost
#当 Apache 服务器引用自己的 URL 时，使用此指定的域名及端口号。与 UseCanonicalName 选
项配合使用
#ServerName www.example.com:80
#是否使用客户端提供的主机名及端口号，如果值为“On”，意味着使用 ServerName 提供的域名
和端口号
UseCanonicalName Off
#设置主服务器的跟文档路径
DocumentRoot "/var/www/html"

#设置根目录的访问控制权限
```

```
<Directory />
   Options FollowSymLinks          #允许符号链接跟踪，访问不在本目录下的文件
   AllowOverride None              #不允许使用目录中.htaccess 文件的配置内容，
即不被它覆盖，.htaccess 文件名称在 AccessFileName 选项中设置
</Directory>

#设置主服务器主目录的访问控制权限，目录位置由 DocumentRoot 选项的内容设置
<Directory "/var/www/html">
 Options Indexes FollowSymLinks
               #Indexes 表示当在目录中找不到指定的文件时，就生成当前目录的文件列表
               # FollowSymLinks 表示允许符号链接跟踪，访问不在本目录下的文件
 AllowOverride None
               #不允许使用目录中.htaccess 文件的配置内容，即不被它覆盖
 Order allow,deny
               #访问规则先执行允许(allow)操作，再执行拒绝(deny)操作
 Allow from all                    #设置 Allow 访问规则，允许所有连接
</Directory>

#不允许用户的个人服务器
<IfModule mod_userdir.c>
   UserDir disable
</IfModule>
#指定主服务器主页文件的名称列表，当客户端访问主服务器时，将依次查找列表中的各个文件
DirectoryIndex index.html index.html.var
#指定.htaccess 配置文件的名称，这个文件是对本目录访问权限进行设置的文件。
AccessFileName .htaccess

#拒绝客户端访问以.ht 开头的文件，即保护访问权限文件
<Files ~ "^\.ht">
   Order allow,deny
   Deny from all
</Files>

#指定负责处理 MIME 文件配置文件的存储位置
TypesConfig /etc/mime.types
#指定默认的 MIME 文件类型为纯文本(或者 HTML 文件)
DefaultType text/plain

#当 mod_mime_magic.c 模块被加载时，指定 Magic 信息配置文件的存放位置
<IfModule mod_mime_magic.c>
#   MIMEMagicFile /usr/share/magic.mime
   MIMEMagicFile conf/magic
</IfModule>

#日志中只记录连接 Apache 服务器的客户端 IP 地址，不记录主机名
HostnameLookups Off
#分发文件时是否启用内存映射功能，对于大内存的服务器来说，建议启用
#EnableMMAP off
#分发文件时是否启动 Sendfile 内核支持。默认为启动状态，如果使用 NFS 文件系统需要禁用此选项。
#EnableSendfile off
#设置错误日志的保存位置
ErrorLog logs/error_log
指定记录错误记录的内容级别
LogLevel warn

#定义日志的记录格式
```

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

#指定访问日志的位置以及日志的格式为 common
#CustomLog logs/access_log common

#设置是否启用 referer 日志和 agent 日志
#CustomLog logs/ referer _log referer
#CustomLog logs/agent_log agent

#指定访问日志的位置以及日志的格式为 combined
CustomLog logs/access_log combined
#在 Apache 生成的页面中使用 Apache 签名
ServerSignature On
#设置/var/www/icons/目录的虚拟目录
Alias /icons/ "/var/www/icons/"

#设置/var/www/icons/目录的访问权限
<Directory "/var/www/icons">
   Options Indexes MultiViews     # MultiViews 指使用内容协商功能决定被发送网
页的性质
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>

#配置 WebDAV 模块，WebDAV(Web-based Distributed Authoring and Versioning)是
基于 HTTP/1.1 的一个通信协议，它为 HTTP/1.1 提供了若干扩展功能，使得应用程序可以直接
将文件写入 Web 服务器中，并且具有在写文件时可以对文件进行加锁，写完文件后再解锁的功能，
还可以支持对文件的版本控制。这个模块极大地增加了 Web 作为一种创作媒体的价值。基于 WebDAV
可以实现一些功能强大的内容管理系统或者配置管理系统
<IfModule mod_dav_fs.c>
   # Location of the WebDAV lock database.
   DAVLockDB /var/lib/dav/lockdb        #指定 DAV 加锁数据库文件的位置
</IfModule>

#设置脚本目录 CGI 的访问别名
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"

#设置 CGI 目录的访问权限
<Directory "/var/www/cgi-bin">
   AllowOverride None
   Options None
   Order allow,deny
   Allow from all
</Directory>

#重定向链接
# Redirect permanent /foo http://www.example.com/bar

#设置自动生成目录列表的显示方式
# FancyIndexing：对每种类型的文件加上一个小图标以示区别
# VersionSort：对同一个软件的多个版本进行排序
# HTMLTable：与 FancyIndexing 一起使用，构建一个简单的 HTML 表格
```

```
IndexOptions FancyIndexing VersionSort NameWidth=* HTMLTable

#当配置了 IndexOptions FancyIndexing 之后，配置下面的选项，用来告知服务器在遇到不同的文件类型或者扩展名时该采用的 MIME 编码格式识别文件类型并显示相应的图标
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*

#当配置了 IndexOptions FancyIndexing 之后，配置下面的选项，用来告知服务器在遇到不同的文件类型或者扩展名时该采用的格式并显示相应的图标，此处有众多的类型识别信息，在此省略了大部分
AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
…

#碰到无法识别的文件时显示此处定义的图标
DefaultIcon /icons/unknown.gif

#为某些类型的文件加入解释文本
#AddDescription "GZIP compressed document" .gz
#AddDescription "tar archive" .tar
#AddDescription "GZIP compressed tar archive" .tgz
#当服务器自动列出目录列表时，在所有生成的页面后附加 README.html 中的内容，在页面的前面附加 HEADER.html 的内容
ReadmeName README.html
HeaderName HEADER.html

#当服务器自动列出目录列表时，下面的这些文件不会列出
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t

#设置网页内容的语言种类()需要浏览器启用内容协商
# DefaultLanguage nl        #设置一种默认语言，没有指定语言的页面采用该语言

#加入对各种语言的支持，由于往往加入的语言种类过多，这里不一一列举
AddLanguage ca .ca
AddLanguage cs .cz .cs
AddLanguage da .dk
…

#当启用内容协商时，设置语言的先后顺序
LanguagePriority en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl pt pt-BR ru sv zh-CN zh-TW
#当有多种语言匹配时，使用 LanguagePriority 列表的第一个匹配，当没有语言可匹配时，使用 LanguagePriority 列表的第一项
ForceLanguagePriority Prefer Fallback

##设置默认字符集为 UTF-8
AddDefaultCharset UTF-8

#添加新的 MIME 类型，会覆盖掉/etc/mime.types 中的设定
#AddType application/x-tar .tgz
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
```

```
#添加支持的压缩编码格式
#AddEncoding x-compress .Z
#AddEncoding x-gzip .gz .tgz

#设定对特定扩展名文件的处理方式
#AddHandler cgi-script .cgi     #将“.cgi”扩展名的文件当做脚本处理(需要其他选
项的支持)
#AddHandler send-as-is asis
AddHandler type-map var

#设定.shtml 文件的类型为 text/html
AddType text/html .shtml
#服务器处理响应时，将.shtml 文件映射到过滤器 INCLUDES 中
AddOutputFilter INCLUDES .shtml

#指定错误响应代码的解释文本
#ErrorDocument 500 "The server made a boo boo."    #以普通文本为内容
#ErrorDocument 404 /missing.html                    #以网页为内容
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"   #以脚本的执行结果作为内容
#ErrorDocument 402 http://www.example.com/subscription_info.html
#重定向到其他 URL

#设置错误页面目录的虚拟目录
Alias /error/ "/var/www/error/"

#加载了 mod_negotiation 和 mod_include 模块的设置
<IfModule mod_negotiation.c>
<IfModule mod_include.c>
   <Directory "/var/www/error">      #设置/var/www/error 目录的访问权限
      AllowOverride None
      Options IncludesNoExec
      AddOutputFilter Includes html
      AddHandler type-map var
      Order allow,deny
      Allow from all
      LanguagePriority en es de fr
      ForceLanguagePriority Prefer Fallback
   </Directory>

#设定发生对应错误代码的显示内容，错误代码众多，在此不一一列举
#    ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var
#    ErrorDocument 401 /error/HTTP_UNAUTHORIZED.html.var
#    ErrorDocument 403 /error/HTTP_FORBIDDEN.html.var
…
</IfModule>
</IfModule>

#对于特定的浏览器给予特定的响应
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0

#解决某些浏览器 BUG 引起的问题
BrowserMatch "Microsoft Data Access Internet Publishing Provider"
redirect-carefully
```

```
BrowserMatch "MS FrontPage" redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
BrowserMatch "^WebDAVFS/1.[0123]" redirect-carefully
BrowserMatch "^gnome-vfs/1.0" redirect-carefully
BrowserMatch "^XML Spy" redirect-carefully
BrowserMatch "^Dreamweaver-WebDAV-SCM1" redirect-carefully

#允许由 mod_status 模块产生状态报告
#<Location /server-status>
#    SetHandler server-status
#    Order deny,allow
#    Deny from all
#    Allow from .example.com
#</Location>

#mod_info 加载时，设置远程服务器配置报告功能
#<Location /server-info>
#    SetHandler server-info
#    Order deny,allow
#    Deny from all
#    Allow from .example.com
#</Location>

#启动代理服务器功能
#<IfModule mod_proxy.c>
#ProxyRequests On
#
#<Proxy *>
#    Order deny,allow
#    Deny from all
#    Allow from .example.com
#</Proxy>

#开启代理服务器处理:Via 头域的功能
#ProxyVia On

#设置代理服务器启动 Cache 功能
#<IfModule mod_disk_cache.c>
#   CacheEnable disk /
#   CacheRoot "/var/cache/mod_proxy"
#</IfModule>
#
#</IfModule>
# End of proxy directives.
```

提示： 关于配置文件 httpd.conf 中的第三部分“虚拟主机”的配置，由于涉及的知识较多，会在下面单独拿出一节来介绍。

8.6 虚拟主机的配置

虚拟主机的出现源于网站的迅猛发展，在“每站一机”的传统方式已经不能满足需求时，虚拟主机技术就应运而生了。

8.6.1　虚拟主机的概述

虚拟主机是指在一台服务器运行多个网站，以实现对客户端的透明性。利用虚拟主机技术，可以把一台真正的主机分成许多“虚拟”的主机，从而实现多用户对硬件资源、网络资源共享，大幅度降低了用户的建站成本。每一台虚拟主机都具有完整的 Web 服务器功能。虚拟主机各用户之间也是完全独立的，从外界来看，虚拟主机和独立主机的表现是完全一样的。目前许多企业建立网站都采用租用 ISP 虚拟主机的方法，这是因为虚拟主机具有以下优点。

1. 节约投资

利用"虚拟主机"技术，可以使用户节省大量不必要的开支，包括购置服务器、租用专线及其他硬件设备和安排专业系统管理人员等多方面的费用。

2. 节约维护费用

通过使用“虚拟主机”，可以节约维护费用，一条 128K 专线在国内最基本的费用为每月上千元，同时因为国内许多地区采用双向计费的方式，网站的访问量越高，用户所要支付的费用就越多，而虚拟服务器的价格一年一般仅为数百元。

3. 可以获得专业的维护而无需维护人员与昂贵的电源系统

专业的网络系统供应商往往花费大量的投资用于建设后备电源系统以应付电力故障，并且拥有一批专业的维护人员，这些资源对于普通的企业来说是不必要的支出。

4. 拥有更加稳定的性能

普通的企业级网站往往只通过某一家 ISP 接入，如果这家供应商有故障，则用户必将受到影响。而采取租用"虚拟主机"的方案往往可以借助服务商的多路由获得稳定的性能，因为大多数的虚拟主机服务商所依赖的主干网一般不止一条，可以保证系统不受某一家供应商的影响。

虚拟主机的实现方式有两种：基于 IP 的虚拟主机和基于域名的虚拟主机。Apache 是率先支持基于 IP 的虚拟主机的服务器之一，自 1.3 版本后，Apache 对两种虚拟主机都提供了支持。

8.6.2　基于 IP 的虚拟主机

在基于 IP 的虚拟主机中，每个网络接口的 IP 地址对应着一台虚拟主机，此时，需要为每一台虚拟主机分配一个独立的 IP 地址。基于 IP 地址的虚拟主机在服务器里绑定多个 IP，然后配置 Apache，把多个网站绑定在不同的 IP 上，访问服务器上不同的 IP，就看到不同的网站。

例如，假设服务器分配了 192.168.1.101 和 192.168.1.102 两个 IP 地址，现需使用这两个 IP 地址分别创建两台虚拟主机，每台虚拟主机都对应不同的主目录，我们需要在主配置文件 httpd.conf 中添加以下语句实现。

```
<VirtualHost 192.168.1.101>        #配置虚拟主机使用的 IP 地址
ServerName 192.168.1.101:80
ServerAdmin web1@test.edu
DocumentRoot "/usr/www/web1"        #虚拟主机根目录
DirectoryIndex index.html
ErrorLog logs/web1/error_log        #虚拟主机日志保存位置
CustomLog logs/web1/access_log combined     #设置日志类型
</VirtualHost>

<VirtualHost 192.168.1.102>
ServerName 192.168.1.102:80
ServerAdmin web2@test.edu
DocumentRoot "/usr/www/web2"
DirectoryIndex default.html
ErrorLog logs/web2/error_log
CustomLog logs/web2/access_log combined
</VirtualHost>
```

创建虚拟主机需要在 Apache 的主配置文件 httpd.conf 中使用<VirtualHost>和</VirtualHost>这对语句进行设置，这对语句必须成对出现，它们之间封装了设置虚拟主机属性的选项。虚拟主机与配置独立的 Web 服务器类似，因此大部分的配置选项都能用在<VirtualHost>和</VirtualHost>语句之间。

上例中的语句<VirtualHost 虚拟主机的 IP>是指明这台虚拟主机使用哪个 IP 地址。如果虚拟主机需要独立的日志文件，应保证日志文件的路径存在，否则 Apache 将不能启动。

8.6.3 基于域名的虚拟主机

基于域名的虚拟主机只需服务器有一个 IP 地址即可创建多台虚拟主机，所有的虚拟主机共享同一个 IP 地址，各虚拟主机之间通过域名进行区分。因为 HTTP 协议访问请求里包含域名信息，所以当 Web 服务器收到访问请求时，就可以根据不同的域名来访问不同的网站。它的优势就是不需要更多的 IP 地址，容易配置。

要建立基于域名的虚拟主机，首先要更改 DNS 服务器的配置，在 DNS 服务器中建立多个此服务器地址的资源记录，使它们解析到相同 IP 地址。例如：

```
www1.test.edu.  IN   A  192.168.1.101
www2.test.edu.  IN   A  192.168.1.101
```

然后，还需要使用这两个域名分别创建两台虚拟主机，每台虚拟主机都对应不同的主目录，在主配置文件 httpd.conf 中添加以下语句实现：

```
NameVirtualHost 192.168.1.101
<VirtualHost www1.test.edu >
ServerName www1.test.edu:80
ServerAdmin web@test.edu
DocumentRoot "/usr/www/web1"
DirectoryIndex index.html
ErrorLog logs/web1/error_log
CustomLog logs/web1/access_log combined
</VirtualHost>
<VirtualHost www2.test.edu >
ServerName www2.test.edu:80
ServerAdmin web@test.edu
```

```
DocumentRoot "/usr/www/web2"
DirectoryIndex default.html
ErrorLog logs/web2/error_log
CustomLog logs/web2/access_log combined
</VirtualHost>
```

创建基于域名的虚拟主机时，必须先用 NameVirtualHost 指令指定哪个 IP 地址负责响应对虚拟主机的请求，然后<VirtualHost 虚拟主机的域名>来指明这台虚拟主机使用哪个域名。

提示： 没有必要为每个虚拟主机指定所有的配置语句，因为虚拟主机中没有指定的配置语句将使用主服务器主配置文档中的配置。

8.7　Web 发布及访问控制

在前面的内容中，我们已经对 Apache 的架设和基本配置做了详细的介绍，但仅仅使用以上的功能还不能很好地满足 Web 服务器的要求，在本节的内容中，我们将对虚拟目录的使用以及用户访问权限的控制做详细的介绍。

8.7.1　创建虚拟目录

要从主目录以外的其他目录中进行发布，就必须创建虚拟目录。虚拟目录是一个位于 Apache 的主目录外的目录，它不包含在 Apache 的主目录中，但在访问 Web 站点的用户看来，它与位于主目录中的子目录是一样的。每个虚拟目录都有一个别名，用户 Web 浏览器中可以通过此别名来访问虚拟目录，例如 http://服务器 IP 地址/别名/文件名，就可以访问虚拟目录下面的任何文件了。使用虚拟目录有以下优点。

1. 便于访问

由于虚拟目录名(别名)通常要比真实目录的路径名短，因此使用虚拟目录名(别名)访问简短、方便。

2. 便于移动站点中的目录

只要虚拟目录名(别名)不变，即使更改了虚拟目录的实际存放位置，无需更改目录的 URL，也不会影响用户的访问。

3. 能灵活加大磁盘空间

虚拟目录能够提供的磁盘空间几乎是无限的。适合于提供对磁盘空间要求加大的 VOD 服务、个人主页服务或其他 Web 服务。

4. 安全性好

由于每个虚拟目录都可以分别设置不同的访问权限，因此非常适合于不同用户对不同目录拥有不同权限的情况。此外，虚拟目录名(别名)通常只有该用户知道，其他不知道虚拟目录名的用户无法访问。黑客也不知道虚拟目录的实际存放位置，难以进行破坏。

使用 Alias 选项可以创建虚拟目录。在主配置文件中，Apache 默认已经创建了两个虚拟目录。这两条语句分别建立了"/icons/"和"/manual"两个虚拟目录，它们对应的物理路径分别是"/var/www/icons/"和"/var/www/error/"。

```
Alias /icons/ "/var/www/icons/"
Alias /error/ "/var/www/error/"
```

例如，我们要创建名为/down 的虚拟目录，它对应的物理路径是"/software/download"，可以使用以下的命令：

```
Alias /down "/software /download"
```

8.7.2 目录权限配置

目录权限配置是指对文件系统图中的目录进行权限设置，指定哪些客户端可以访问该目录。对于可以访问的客户端，还能够指定客户端在该目录中可以进行的操作，例如列出目录内容、执行等。在 Apache 中配置目录访问控制有两种方法：既可以在主配置文件 httpd.conf 中配置目录访问控制选项，而针对每一个目录的访问控制，用户还可以在相应的目录中创建".htacess"的文件，此文件同样可以作为目录访问控制的配置信息。

1. 通过 httpd.conf 配置目录访问控制

我们可以在 httpd.conf 配置文件中使用<Directory 目录路径>和</Directory>这对语句为主目录或虚拟目录设置权限，它们是一对容器语句，必须成对出现，它们之间封装的是具体的设置目录权限选项，这些选项仅对被设置目录及其子目录起作用。下面是主配置文件中设置目录权限的例子。

```
<Directory "/var/www/icons">
Options Indexes MultiViews
AllowOverride None
Order allow, deny
Allow from all
</Directory>
```

从上面的例子我们可以看出，配置目录访问控制选项的格式如下：

```
<Directory 目录路径>
[访问控制选项]
</Directory>
```

其中，访问控制选项主要有以下几类：

- 授权访问选项(AuthConfig)：包括 AuthDBMGroupFile、AuthDBMUserFile、AuthGroupFile、AuthName、AuthType、AuthUserFile 和 Require 等。
- 文件控制类选项(FileInfo)：包括 AddLanguage、AddType、DefaultType、ErrorDocument、LanguagePriority、AddHander 和 AddOutputFilter 等。
- 目录显示方式类选项(Indexes)：包括 AddDescription、AddIcon、AddIconByEncoding、AddIconByType、DefaultIcon、FancyIndexing、HeaderName、IndexIgnore、IndexOptions 和 ReadmeName 等。

- 客户端访问控制类选项(Limit)：包括 Allow、Deny 和 Order 等。
- 目录访问控制类选项(Options)：包括 Options 和 XbitHack 等。

提示： 由于控制目录访问的选项较多，篇幅所限，在这里不对这些选项进行一一的解释，有兴趣的读者可以参考 Apache 的 manual 来学习这些选项的具体含义和使用方法。

其中，Options 选项用于定义目录使用哪些特性，包括 Indexes、MultiViews 和 ExecCGI 等，是应用非常广泛的选项，它可以使用的选项及功能如下：

- All：All 包含了除 MultiViews 之外的所有特性，如果没有 Options 语句，默认为 All。
- None：禁止所有功能。
- MultiViews：允许内容协商的多重视图。
- Indexes：表示如果该目录下无 index 文件，那么允许显示该目录下的文件列表。
- IncludesNoExec：允许服务器端包含功能，但禁用执行 CGI 脚本。
- Includes：允许目录浏览。
- FollowSymLink：可以在该目录中使用符号连接。
- SymLinksIfOwnerMatch：在该目录中仅仅跟踪本站点内的链接。
- ExecCGI：允许在该目录下执行 CGI 脚本。

提示： MultiViews 是 Apache 的一个智能特性。当客户访问目录中一个不存在的对象时，如访问“http://192.168.16.177/icons/a”，则 Apache 会查找这个目录下所有 a.*文件。由于 icons 目录下存在 a.gif 文件，因此 Apache 会将 a.gif 文件返回给客户，而不是返回出错信息。

2. 使用.htaccess 文件设置目录访问权限

每一个目录中都可以包含一个.htaccess 文件，Apache 服务器可以读取该文件的内容作为目录访问控制的配置，使用 AllowOverride 可以指定哪些选项可以被.htaccess 文件的内容覆盖。如果设置为 None，那么服务器将忽略.htaccess 文件，如果设置为 All，那么所有在.htaccess 文件中的选项都会被采用，并且将覆盖主配置文件中的相应配置。

3. 客户端访问控制指令

客户端访问控制类指令也是使用非常广泛的一类指令。前面我们讲到，客户端访问控制类指令一共有三个：

- Order：用于指定执行允许和拒绝访问规则的先后顺序。
- Deny：拒绝访问控制列表。
- Allow：允许访问控制列表。

其中，Order 选项用于定义缺省的访问权限与 Allow 和 Deny 语句的处理顺序。Allow 和 Deny 语句可以针对客户机的域名或 IP 地址进行设置，以决定哪些客户机能够访问服务器。Order 语句通常设置为以下两种值之一。

- allow, deny：缺省禁止所有客户机的访问，且 Allow 语句在 Deny 语句之前被匹

配。如果某条件既匹配 Deny 语句又匹配 Allow 语句，则 Deny 语句会起作用(因为 Deny 语句覆盖了 Allow 语句)。

- deny, allow：缺省允许所有客户机的访问，且 Deny 语句在 Allow 语句之前被匹配。如果某条件既匹配 Deny 语句又匹配 Allow 语句，则 Allow 语句会起作用(因为 Allow 语句覆盖了 Deny 语句)。

下面的例子很好地解释了 Order 顺序的不同所造成的规则执行结果的不同。

```
#仅允许来自网络 192.168.16.0/24 客户机的访问
Order allow, deny
Allow from 192.168.16.0/24

#这两条语句是允许所有客户机的访问
Order deny , allow
Allow from 192.168.16.0/24
```

8.7.3 用户认证

用户认证是指用户通过浏览器访问某些受到保护的资源时，需要提供正确的用户名和密码才能访问。用户认证在网络安全中是非常重要的技术之一，它是保护网络系统资源的第一道防线。用户认证控制着所有登录并检查访问用户的合法性，其目标是仅让合法用户以合法的权限访问网络系统的资源。当用户第一次访问了启用用户认证目录下的任何文件，浏览器会显示一个对话框，要求输入正确的登录用户名和口令进行用户身份的确认。若是合法用户，则显示所访问的文件内容。此后访问该目录的每个文件时，浏览器会自动送出用户名和密码，不用再次输入，直到关闭浏览器为止。用户认证功能起到了一个屏障的作用，限制非授权用户非法访问一些私有的内容。

在 Apache 中支持两种类型的认证方式：基本认证和摘要认证。基本认证是传统的基于用户名和密码的认证方式。而摘要认证用来提供比基础认证更高级别的安全，它是一种基于挑战-应答模式的认证模型。这是一种常用的技术，用于证明某人知道某个秘密，而不要求他以容易被窃听的明文形式发送该秘密。尽管摘要认证更加安全，但并不是所有的浏览器都支持这种认证。所以大多数情况下使用的依然是基本认证。

下面我们通过一个实例讲解在 Apache 中启用用户认证功能的方法。

假设有一个名为 mywebsite 的虚拟目录，其对应的物理路径是/usr/local/mywebsite，现需要对其启用用户认证功能，只允许用户名为 test 的用户访问，设置用户认证的具体操作步骤如下。

1) 建立密码文件

要实现用户认证功能，首先要建立用户名和密码的文件。Apache 自带的 htpasswd 命令提供了建立和更新存储用户名、密码的文本文件的功能。需要注意的是，这个文件必须放在不能被网络访问的位置，以避免被下载。htpasswd 命令的使用格式如下：

```
htpasswd -c <认证密码文件名> <用户名>
```

例如此文件放在/etc/httpd/目录下，文件名为 passwd。使用以下命令建立口令文件。

```
[root@CentOS ~]# htpasswd -c /etc/httpd/passwd test
New password:
```

```
Re-type new password:
Adding password for user test
```

其中，“-c”选项表示无论密码文件是否已经存在，都会重新写入文件并删去原有内容。

2) 建立虚拟目录并配置用户认证

在 Apache 的主配置文件 httpd.conf 中加入以下语句建立虚拟目录并配置用户认证。Apache 中配置用户认证的主要选项如表 8-3 所示。

表 8-3　Apache 用户认证选项及其含义

选　项	格　式	含　义
AuthName	AuthName 认证名称	定义受保护领域的名称
AuthType	AuthType Basic|Digest	定义使用基本认证还是摘要认证
AuthGroupFile	AuthGroupFile 文件名	指定认证组文件的位置
AuthUserFile	AuthUserFile 文件名	指定认证文件的位置

例如下面的例子：

```
Alias /mywebsite "/usr/local/mywebsite"
<Directory "/usr/local/ mywebsite ">
AuthType Basic
AuthName "This is a private directory. Please Login:"
AuthUserFile /etc/httpd/passwd
Require user test
</Directory>
```

上面的例子设置的主要内容有：

(1) 设置认证类型 AuthType Basic

AuthType 选项定义了对用户实施认证的类型，最常用的是由 mod_auth 提供的 Basic。

(2) 设置认证领域内容 AuthName "This is a private directory. Please Login:"

AuthName 选项定义了 Web 浏览器显示输入用户/密码对话框时的领域内容。

(3) 设置密码文件的路径 AuthUserFile /etc/httpd/passwd

AuthUserFile 选项定义了口令文件的路径，即使用 htpasswd 建立的口令文件。

(4) 设置允许访问的用户 Require user test

Require user 选项定义了允许哪些用户访问，各用户之间用空格分开。

3) 测试用户认证

进行完以上的设置后，下面我们来测试一下访问认证网站的结果。

首选需要在服务器中使用命令“/etc/init.d/httpd restart”来重启 Web 服务。

其次，测试用户认证时最好在虚拟目录中建立一个名为 index.html 的文件，否则输入正确的用户名和口令后，由于虚拟目录既没有设置默认文档，也没有设置允许目录浏览，所以会出现“403 Forbidden”的错误信息。另外，我们还要使 Apache 有访问测试目录和文件的权限。使用以下的命令。

```
[root@CentOS ~]# chown apache /usr/local/mywebsite/
[root@CentOS ~]# chown apache /usr/local/mywebsite/index.html
```

```
[root@CentOS ~]# chmod 700  /usr/local/mywebsite/index.html
[root@CentOS ~]# chmod 700  /usr/local/mywebsite/
```

在客户端的 Web 浏览器中访问这个虚拟目录，这时 Web 浏览器会弹出输入用户名和密码的对话框，如图 8-6 所示。

输入正确的用户名和密码，就能访问该网站内容，如图 8-7 所示。

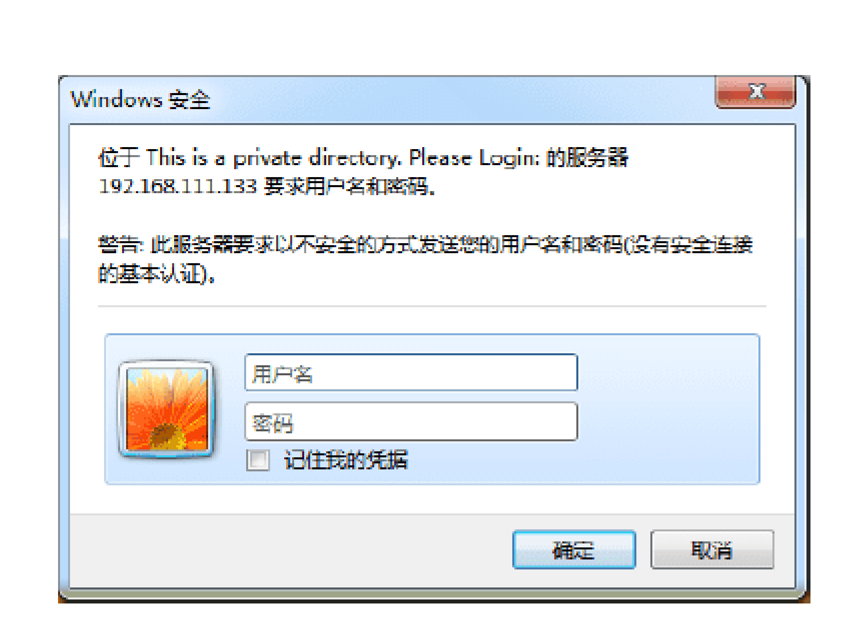



图 8-6　输入用户名和密码的对话框

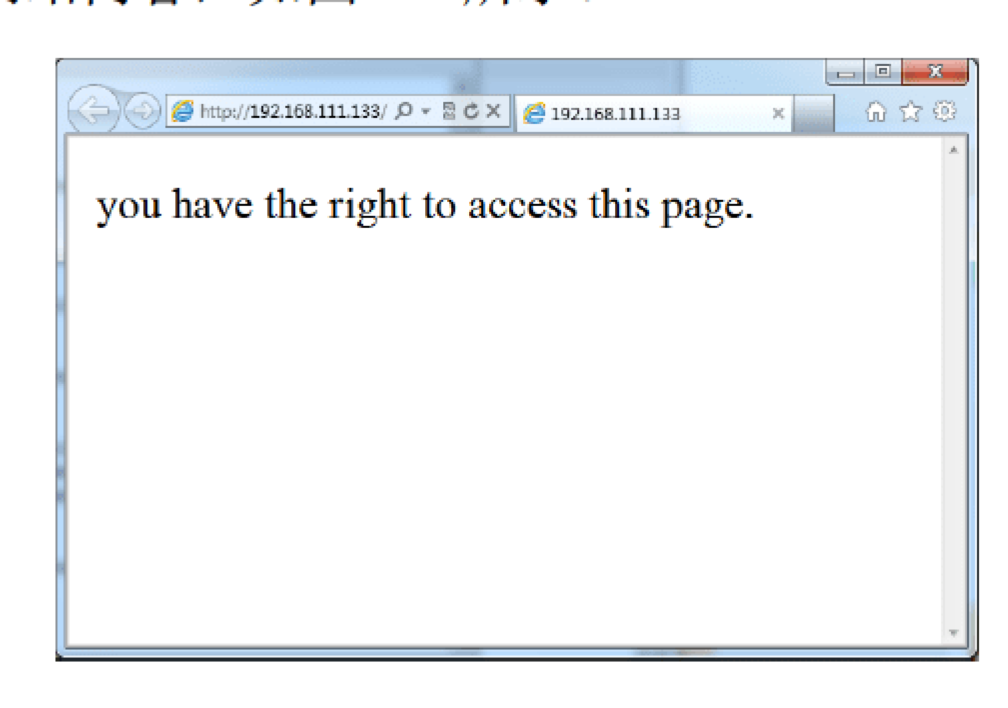


图 8-7　经过认证后可以正常访问

如果输入的用户名和密码不正确，则出现“Authorization Required”的错误信息，如图 8-8 所示。

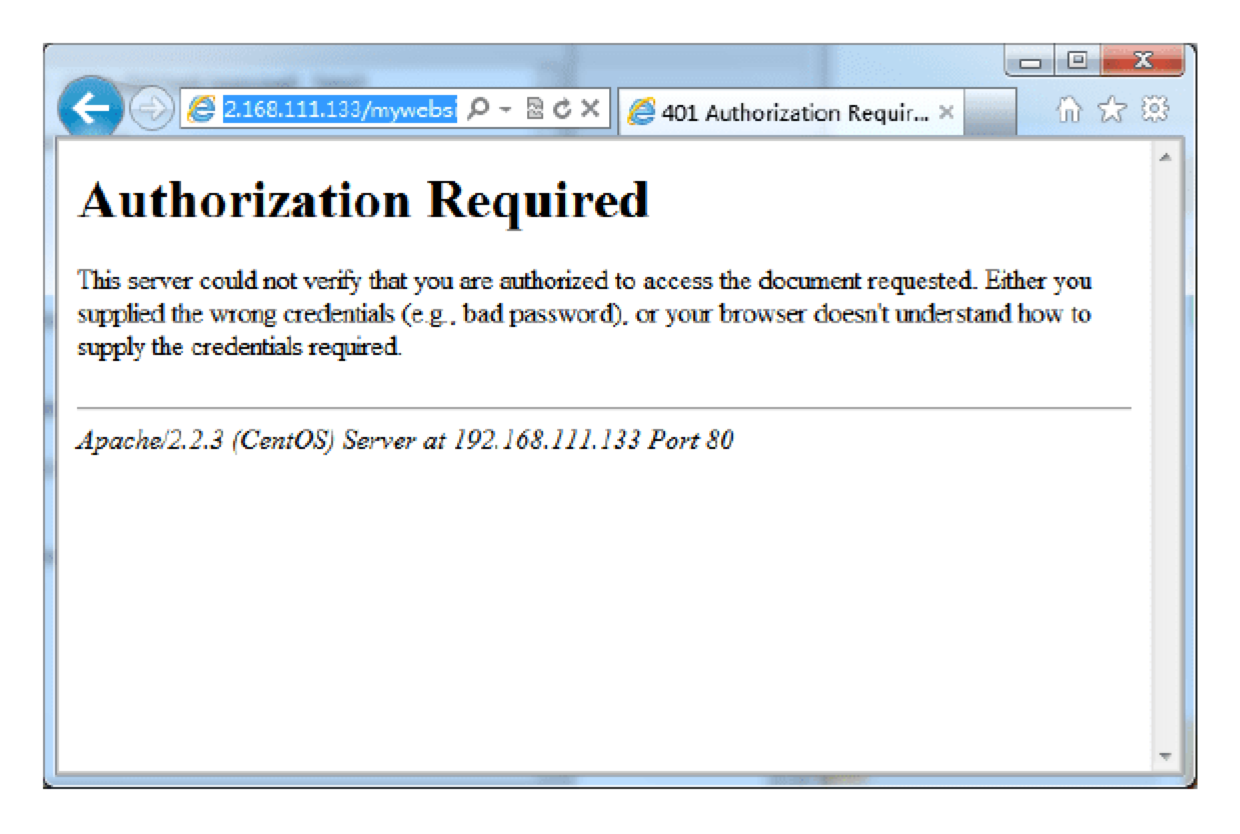


图 8-8　用户认证失败的错误信息

上述的认证方法可以使用.htaccess 文件来实现，其效果是一样的。

8.8　配置 Apache 支持动态网页

除了对静态 HTML 文件的支持，Apache 服务器还可以支持多种形式的动态网页，包括 CGI 脚本、PHP 以及 JSP 语言等。下面我们来详细介绍如何配置 Apache 服务器来支持这些语言。

8.8.1　CGI 运行环境的配置

CGI(Common Gateway Interface，通用网关接口)用于连接网页和 Web 服务器应用程序的接口。HTML 语言的功能比较贫乏，难以完成诸如访问数据库等一类的操作，而实际的

情况则是经常需要先对数据库进行操作(比如文件检索系统)，然后把访问的结果动态地显示在网页上。诸如此类的需求只用 HTML 是无法做到的，所以 CGI 便应运而生。CGI 是在 Web 服务器运行的一个可执行程序，由网页的一个超链接激活进行调用，并对该程序的返回结果进行处理，显示在客户端的 Web 浏览器上。用 CGI 程序可以实现处理网页的表单处理、数据库查询、发送电子邮件等工作。CGI 使网页变得不再是静态的，而是交互式的。Web 服务器的 CGI 程序需要有用户调用才会执行，浏览器、Web 服务器和 CGI 程序之间的基本工作流程如下：

(1) 用户通过 Web 浏览器访问 CGI 程序。

(2) Web 服务器接收用户请求并交给 CGI 程序处理。

(3) CGI 程序执行基于输入数据的操作，包括查询数据库、计算数值或调用系统中其他程序。

(4) CGI 程序产生某种 Web 服务器能理解的输出结果。

(5) Web 服务器接收来自 CGI 程序的输出并且把它传回 Web 浏览器。

要让 CGI 程序能够正常运行，首先必须在 Linux 中安装 Perl 语言解析器，使 Linux 能够正常运行 Perl 语言编写的脚本，其次，需要配置 httpd.conf 文件使 Apahce 支持 CGI 脚本。具体的操作步骤如下。

1. Perl 语言解释器的安装

CGI 可以用任何一种语言编写，只要这种语言具有标准输入、输出和环境变量，如 Perl、C、C++、Java。其中，Perl 易编译调试、移植性颇强，在众多的 CGI 编程语言中，Perl 以其良好的支持、容易上手等特点脱颖而出，几乎成为了 CGI 的标准语言。每当人们提到 CGI 的时候，必然会想到 Perl。Perl(Practical Extraction and Reporting Language，实用摘录与报告语言)，自 1987 年初次登台亮相以来，它的用户数一直急剧膨胀。Perl 不是由某个公司大力推广而得到发展的，正如 Java 那样，Perl 全凭自身的优势来发展。从最初被当作一种在跨平台环境中书写可移植工具的高级语言开始，Perl 就已经被广泛地认为是一种工业级的强大工具。Perl 特别适合系统管理和 Web 编程。Perl 实际上已经被所有 Linux(包括 UNIX)捆绑在一起作为标准部件发布了，如今的 Perl 语言已经被移植到除了 Linux 之外的多种操作平台上。

默认情况下，大多数 Linux 的发行版本都已经将 Perl 语言解释器安装在系统上，读者可使用下面的命令检查系统是否已经安装了 Perl 解释器或查看已经安装了何种版本。

```
[root@CentOS ~]# rpm -q perl
perl-5.8.8-27.el5
```

从命令执行结果看出，在 CentOS 5.5 中，Perl 解释器已安装，它的版本为 5.8.8-27。

如果系统还没有安装 Perl 解释器，应将 CentOS 的安装光盘放入光驱，加载光驱后在光盘的 Server 目录下找到 Perl 解释器的 RPM 安装包文件 perl-5.8.8-27.el5.i386.rpm，使用下面命令安装 Perl 解释器。

```
[root@CentOS ~]# rpm -ivh /mnt/cdrom/CentOS/perl-5.8.8-27.el5.i386.rpm
```

2. httpd.conf 文件的配置

首选需要在设置存放 CGI 文件的目录权限，设置存放 CGI 文件的目录权限可以告诉

Apache 允许 CGI 程序在哪些目录下运行。使用 Options 选项指定允许运行 CGI 执行的目录的格式如下：

```
<Directory "/var/www/hrdocs/somedir">
Options +ExecCGI
</Directory>
```

另外，也可以使用.htaccess 文件来实现上述功能，即将格式中的“Options +ExecCGI”改为“AllowOverride Options”，然后在相应目录中建立.htaccess 文件，将“Options +ExecCGI”放入该文件中，可以达到同样的效果。

其次，需要标明 CGI 程序的文件类型，在 httpd.conf 中添加以下选项。

```
AddHandler cgi-script .cgi pl
```

该语句告诉 Apache 扩展名为“.cgi”的文件是 CGI 程序。“pl”表示同时想运行扩展名为.pl 的文件。

进行以上的设置后，就完成了 Apache 服务器支持 CGI 脚本的设置，下面我们同一个简单的测试来验证 CGI 运行环境。

我们将/var/www/html 目录作为执行 CGI 脚本的目录，即在 httpd.conf 中添加以下选项。

```
<Directory "/var/www/html">
Options +ExecCGI
</Directory>
```

在 CGI 文件存放的目录中建立一个名为 test.cgi 的文件，该文件的内容如下。

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "<h1>Hello World!</h1>\n";
```

这个脚本程序的含义是使用 print 语句输出两行字符串，其中“!/usr/bin/perl”是每一个 Perl 都必须有的，它告诉操作系统这是一个 Perl 的脚本程序。下面，我们将此文件设为 Apache 可执行的文件，使用以下的命令：

```
[root@CentOS ~]# chmod a+x /var/www/html/test.cgi
```

此时我们可以直接在操作系统中执行 test.cgi 来查看输出效果，如下面的代码所示：

```
[root@CentOS html]# ./test.cgi
Content-type: text/html
<h1>Hello World!</h1>
```

现在再用客户端的浏览器访问“http://Linux 服务器的 IP 地址/test.cgi”，如果出现如图 8-9 所示的“Hello World!”，则代表 CGI 运行环境配置成功。

CGI 程序的功能非常强大，从简单的文本显示到处理用户提交的数据都可以实现，具体的实现方法有兴趣的读者可以参考相关资料。

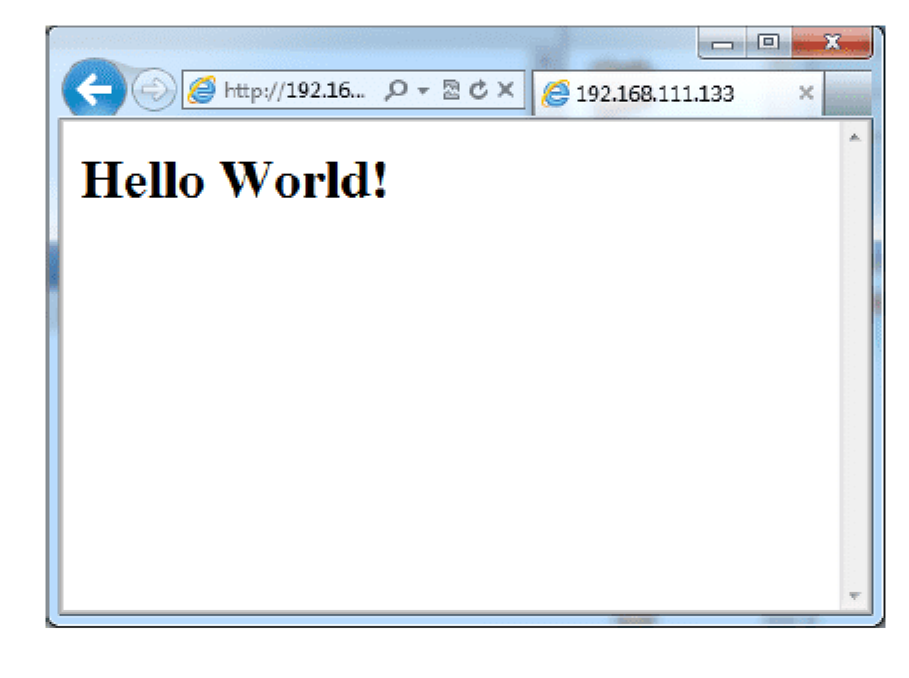


图 8-9 程序运行结果

8.8.2　PHP 运行环境的设置

PHP(PHP Hypertext Preprocessor，超级文本预处理语言)是一种 HTML 内嵌式的语言，PHP 与微软公司的 ASP 的用法非常相似，都是一种在服务器端执行的“嵌入 HTML 文档的脚本语言”，语言的风格类似于 C 语言，现在被很多的网站编程人员广泛运用。PHP 独特的语法混合了 C、Java、Perl 以及 PHP 自有的语法。它可以比 CGI 或者 Perl 更快速地执行动态网页。用 PHP 做出的动态页面与其他的编程语言相比，PHP 是将程序嵌入到 HTML 文档中去执行，执行效率比完全生成 HTML 标记的 CGI 要高许多；与同样是嵌入 HTML 文档的脚本语言 JavaScript 相比，PHP 在服务器端执行，充分利用了服务器的性能；PHP 执行引擎还会将用户经常访问的 PHP 程序驻留在内存中，其他用户在访问这个程序时就不需要重新编译程序了，只要直接执行内存中的代码就可以了，这也是 PHP 高效率的体现之一。PHP 具有非常强大的功能，所有的 CGI 或者 JavaScript 的功能 PHP 都能实现，而且支持几乎所有流行的数据库和操作系统。PHP 具有下列特点。

- 跨平台：PHP 程序可以运行在 UNIX、Linux 或 Windows 操作系统下。
- 嵌入 HTML：因为 PHP 语言可以嵌入到 HTML 内部，所以 PHP 很容易学习。
- 简单的语言：与 Java 和 C++不同，PHP 语言坚持以基本语言为基础，然而它的功能强大到足以支持任何类型的 Web 站点。
- 效率高：和其他的解释性语言相比，PHP 系统消耗较少的系统资源。当 PHP 作为 Apache Web 服务器的一部分时，运行代码不需要调外部二进制程序，服务器解释脚本不需要承担任何额外负担。
- 支持各种数据库：用户可以使用 PHP 存取 Oracle、Sybase、MS-SQL、MySQL、PostgreSQL、dBase、FilePro 和 Informix 等类型的数据库。
- 文件存取：PHP 有许多支持文件存取函数。
- 文本处理：PHP 有许多函数处理字符串，其中包括模式匹配的能力。
- 复杂的变量：PHP 支持标量、数组、关联数组等变量，这给用户提供了支持其他的高级数据结构的坚实基础。
- 支持图像处理：用户可以使用 PHP 动态创建图像。

1. PHP 解释器的安装

读者可使用下面的命令检查系统是否已经安装了 PHP 解释器或查看已经安装了何种版本。

```
[root@CentOS /]# rpm -q php
php-5.1.6-27.el5
```

从图中可以看到，系统虽然已经安装了 PHP 包，但安装的并不完全，还需要 php-cli-5.1.6-27.el5.i386.rpm 以及 php-common-5.1.6-27.el5.i386.rpm 这两个程序。我们可以在 CentOS 的安装光盘中找到这两个 RPM 安装包，使用以下命令安装 PHP 解释器。

```
[root@CentOS usr]# rpm -ivh php-common-5.1.6-27.el5.i386.rpm
Preparing...                ########################################### [100%]
[root@CentOS usr]# rpm -ivh php-cli-5.1.6-27.el5.i386.rpm
Preparing...                ########################################### [100%]
```

安装成功后，所有的安装内容包括：

```
[root@CentOS /]# rpm -ql php
/etc/httpd/conf.d/php.conf
/usr/lib/httpd/modules/libphp5-zts.so
/usr/lib/httpd/modules/libphp5.so
/var/lib/php/session
/var/www/icons/php.gif
```

2. 了解 php.conf 文件

在 Apache 主配置文件 httpd.conf 中我们介绍过，“Include conf.d/*.conf”选项的含义是将目录/etc/httpd/conf.d/中的所有*.conf 文件包含到 httpd.conf 中。PHP 解释器的安装程序会自动在目录/etc/httpd/conf.d/中建立一个名为 php.conf 的配置文件，这个文件包含了 PHP 的配置选项。下面我们来介绍 php.conf 文档中的内容。

```
[root@CentOS conf.d]# more php.conf
#由于篇幅所限，在此省略了 php.conf 文件中的所有英文解释内容
<IfModule prefork.c>
  LoadModule php5_module modules/libphp5.so
</IfModule>
<IfModule worker.c>
LoadModule php5_module modules/libphp5-zts.so   #装载 php 模块
</IfModule>
AddHandler php5-script .php                      #添加解析器来处理.php 文件
AddType text/html .php                           #设定.php 文件的媒体类型
DirectoryIndex index.php                         #添加 index.php 为主页文件
```

另外，由于历史原因，许多原来许多基于 PHP3 的程序文件扩展名为.php3。为了能让这些 PHP3 的程序文件运行，应该在 php.conf 文件中为.php3 扩展名的文件建立映射。编辑/etc/httpd/conf.d/php.conf，将“AddHandler php5-script .php”选项改为“AddHandler php5-script .php .php3”。

3. 测试 PHP 运行环境

测试 PHP 运行环境的具体步骤如下。

在 Apache 主目录/var/www/html/中建立一个名为 test.php 的文件，该文件的内容如下。

```
<? phpinfo(); ?>
```

在客户端的浏览器中访问“http://Linux 服务器的 IP 地址/test.php”，如果出现如图 8-10 所示的 PHP 的信息页面，则表示 PHP 运行环境配置成功。



图 8-10 PHP 测试页面

8.8.3 JSP 运行环境的配置

JSP(Java Server Pages)是由 Sun Microsystems 公司倡导、许多公司一起参与建立的一

种基于 Java 技术的动态网页技术标准。在传统的网页 HTML 文件(*.htm、*.html)中嵌入了 Java 代码的一个脚本，由脚本完成查询数据库、重新定向网页和发送电子邮件等动态操作。所有程序操作都在服务器端执行，网络上传送给客户端的仅是得到的 HTML 结果。在这一点上，JSP 与 ASP 和 PHP 等脚本语言一样。但 JSP 与其他脚本不同的是，ASP 和 PHP 等传统脚本语言由服务器直接解释这个脚本，而 JSP 则由 JSP 容器(如 Tomcat)首先将其转化为 Servlet，然后再调用 Javac 编译器将 Servlet 编译为二进制的 Class 文件，服务器最终运行的是 Class 文件，所以运行效率要比传统解析性的脚本语言更高。

自 JSP 推出以来，得到了众多大公司的支持，纷纷推出了支持 JSP 技术的服务器，例如 IBM、Oracle 等，所以 JSP 迅速成为了商业应用的服务器端语言。Apache 和 Tomcat 是 Apache 基金会下属的两个项目，两者能够实现很好的兼容，Apache 作为前端的 HTTP Web 服务器，Tomcat 作为后端的 Servlet 容器。

1. JDK 的安装

由于 Tomcat 需要在 Java 平台上运行，因此，首先需要安装 Java 开发工具，即 JDK。我们可以从 http://www.oracle.com/technetwork/java/javase/downloads/ 网站处下载 JDK 的最新版本，如图 8-11 所示。目前的最新版本为 7u3，文件名为 jdk-7u3-linux-i586.rpm。下载此 RPM 包后，将其放入 Linux 系统中，运行以下命令开始安装。

```
[root@CentOS usr]# rpm -ivh jdk-7u3-linux-i586.rpm
Preparing...            ########################################### [100%]
   1:jdk                ########################################### [100%]
Unpacking JAR files...
        rt.jar...
        jsse.jar...
        charsets.jar...
        tools.jar...
        localedata.jar...
        plugin.jar...
        javaws.jar...
        deploy.jar...
```

Java SE Development Kit 7u3

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86 (32-bit)	63.65 MB	jdk-7u3-linux-i586.rpm
Linux x86 (32-bit)	78.66 MB	jdk-7u3-linux-i586.tar.gz
Linux x64 (64-bit)	64.53 MB	jdk-7u3-linux-x64.rpm
Linux x64 (64-bit)	77.3 MB	jdk-7u3-linux-x64.tar.gz
Solaris x86 (32-bit)	135.96 MB	jdk-7u3-solaris-i586.tar.Z
Solaris x86 (32-bit)	81.4 MB	jdk-7u3-solaris-i586.tar.gz
Solaris SPARC (32-bit)	138.92 MB	jdk-7u3-solaris-sparc.tar.Z
Solaris SPARC (32-bit)	86.07 MB	jdk-7u3-solaris-sparc.tar.gz
Solaris SPARC (64-bit)	16.14 MB	jdk-7u3-solaris-sparcv9.tar.Z
Solaris SPARC (64-bit)	12.31 MB	jdk-7u3-solaris-sparcv9.tar.gz
Solaris x64 (64-bit)	14.46 MB	jdk-7u3-solaris-x64.tar.Z
Solaris x64 (64-bit)	9.25 MB	jdk-7u3-solaris-x64.tar.gz
Windows x86 (32-bit)	84.12 MB	jdk-7u3-windows-i586.exe
Windows x64 (64-bit)	87.41 MB	jdk-7u3-windows-x64.exe

图 8-11　JDK 下载页面

JDK 安装完成后，我们还需要运行以下的命令来设置环境变量，这些命令最好设为开机自动执行。

```
[root@CentOS /]# export JAVA_HOME=/usr/java/jdk1.7.0_03
[root@CentOS /]# export
CLASSPATH=.:$JAVA_HOME/lib.dt.jar:$JAVA_HOME/lib/tools.jar
[root@CentOS /]# export PATH=$PATH:$JAVA_HOME/bin
```

2. 安装 Tomcat

默认情况下 CentOS 安装程序不会将 Tomcat 服务安装在系统上，读者可以使用下面的命令检查系统是否已经安装了 Tomcat 服务或查看已经安装了何种版本。

```
[root@CentOS /]# rpm -q tomcat
package tomcat is not installed
```

从显示内容可以看出，系统当前还没有安装 Tomcat 服务。我们可以在 CentOS 5.5 的安装光盘中找到 Tomcat5 的安装程序，Tomcat 服务的关联程序非常多，需要用户逐个安装这些 RPM 包，在此我们不详细介绍这种安装方法，而是使用一种更加简单的安装方法，在 Tomcat 的官方网站中提供了 Tomcat 的安装程序，并且解压缩后就可以直接使用。读者可以访问网址 http://tomcat.apache.org/来获得最新的安装程序，如图 8-12 所示。

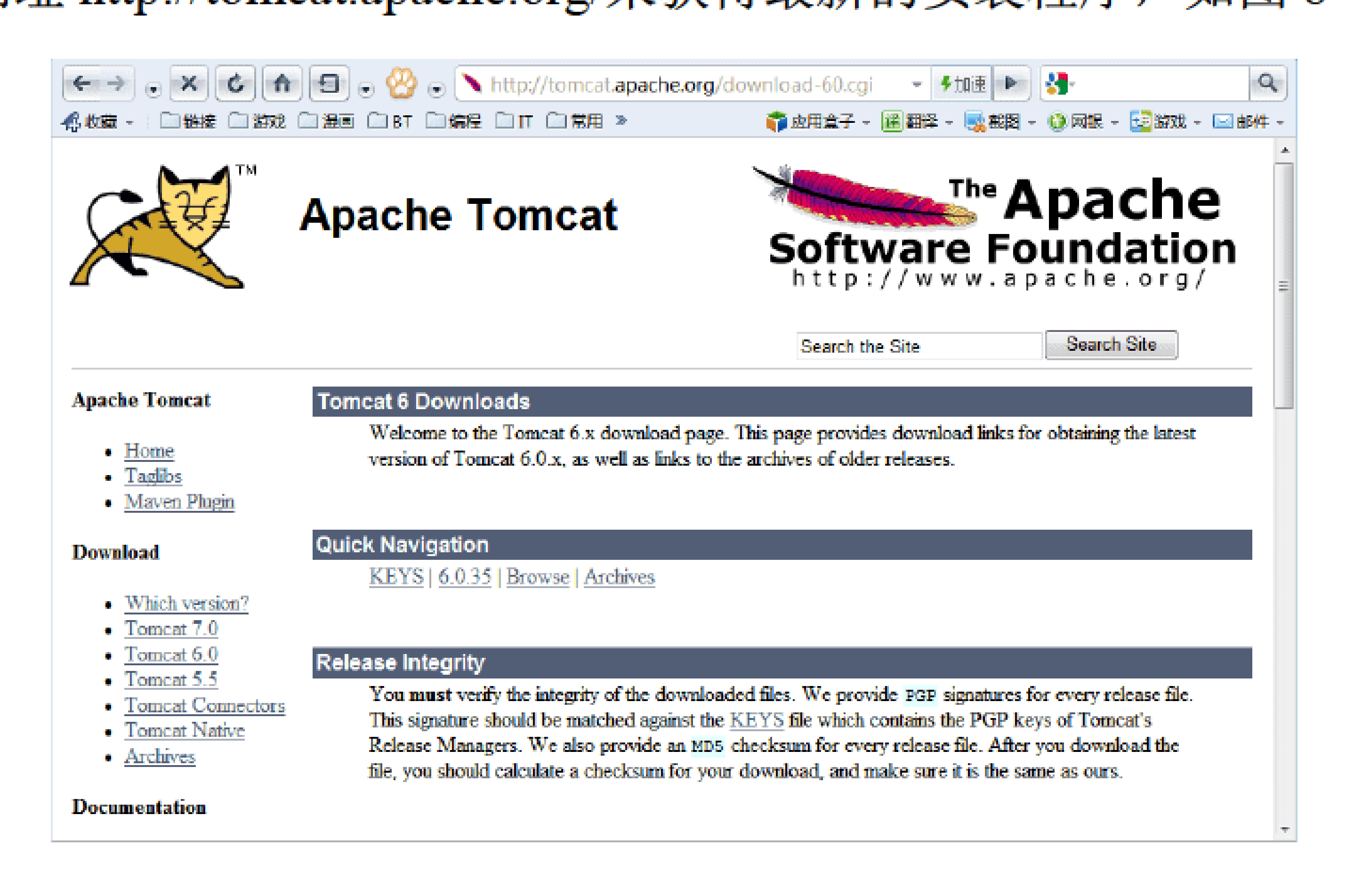


图 8-12　Tomcat 官方网站

Tomcat 最新的版本为 7.0.26，但此处我们以实际使用较多的 6.0.35 版本为例介绍安装过程。

首先将文件 apache-tomcat-6.0.35.tar.gz 复制到当前目录，使用以下命令将其解压缩。

```
[root@CentOS usr]# tar -zvxf ./apache-tomcat-6.0.35.tar.gz
```

解压完成后，所有文件都放在 apache-tomcat-6.0.35 目录中，其默认的配置文件已经可以使用，因此，我们输入以下命令就可以启动 Tomcat 了。

```
[root@CentOS usr]# ./apache-tomcat-6.0.35/bin/startup.sh
Using CATALINA_BASE:   /usr/apache-tomcat-6.0.35
Using CATALINA_HOME:   /usr/apache-tomcat-6.0.35
Using CATALINA_TMPDIR: /usr/apache-tomcat-6.0.35/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /usr/apache-tomcat-6.0.35/bin/bootstrap.jar
```

为了确定 Tomcat 是否正常运行，我们可以使用 ps 命令来查看 Tomcat 进程是否已经启动，如下面的内容所示。

```
[root@CentOS usr]# ps -eaf | grep tomcat
root      7273     1  9 12:00 pts/0    00:00:02 /usr/bin/java -
Djava.util.logging.config.file=/usr/apache-tomcat-
6.0.35/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -
Djava.endorsed.dirs=/usr/apache-tomcat-6.0.35/endorsed -classpath
/usr/apache-tomcat-6.0.35/bin/bootstrap.jar -Dcatalina.base=/usr/apache-
tomcat-6.0.35 -Dcatalina.home=/usr/apache-tomcat-6.0.35 -
Djava.io.tmpdir=/usr/apache-tomcat-6.0.35/temp
org.apache.catalina.startup.Bootstrap start
root      7300  7207  0 12:00 pts/0    00:00:00 grep tomcat
```

在 Tomcat 的进程中之所以有这么多参数，主要是因为 Tomcat 是运行在 Java 虚拟机环境下。默认情况下，Tomcat 会监听 TCP 8080 端口，我们再查看一下此端口是否已经在监听状态。

```
[root@CentOS usr]# netstat -an | grep :8080
tcp      0      0 :::8080      :::*         LISTEN
```

可以看到，端口的监听也处于正常状态，为了保证客户端能够正常访问 Tomcat 服务器，还需要在防火墙开放 8080 端口。

```
iptables -I INPUT -p tcp --dport 8080 -j ACCEPT
```

经过上述的配置以后，我们可以使用客户端访问 Tomcat 服务器了，正常情况下，在浏览器中输入地址“http://Linux 服务器地址:8080”，如果出现如图 8-13 所示的测试页面，则说明客户端已经能够正常访问 Tomcat 服务器了。

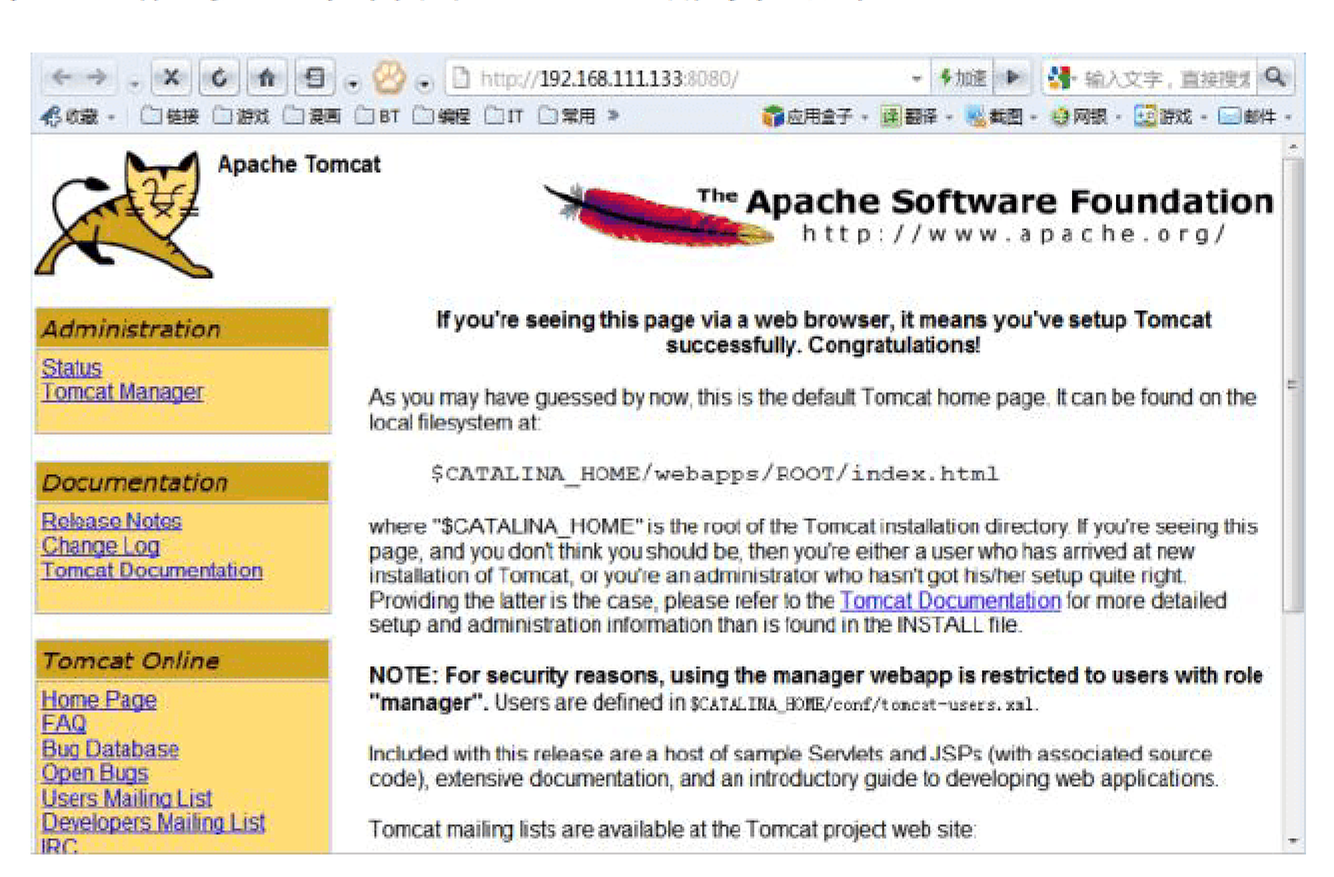


图 8-13　Tomcat 测试页面

3. 整合 Apache 和 Tomcat

虽然 Tomcat 可以独立作为 Web 服务器运行，但 Tomcat 的 Web 功能远没有 Apache 强大，所以在实际应用中是通过 mod_jk 连接器(Connectors)将 Apache 和 Tomcat 整合在一起提供服务的，Apache 处理静态页面的请求，Tomcat 则用于处理 Servlet 和 JSP 程序。

1) 下载 mod_jk

我们以 mod_jk 连接器的稳定版本 1.2.23 为例，介绍使用 mod_jk 的方法，使用 Web 浏览器访问 http://www.apache.org/dist/tomcat/tomcat-connectors/jk/binaries/linux/jk-1.2.23/，

单击页面的“mod_jk-1.2.23-apache- 2.2.x-linux-i686.so”超链接，下载 mod_jk，文件大小约为 589KB，如图 8-14 所示。

archive.apache.org

This site contains the historical archive of old software releases.

For current releases, please visit the mirrors.

Name	Last modified	Size	Description
Parent Directory		-	
mod_jk-1.2.23-apache-2.2.x-linux-i686.so	2007-07-06 02:20	589K	
mod_jk-1.2.23-apache-2.2.x-linux-i686.so.asc	2007-07-06 02:20	189	
mod_jk-1.2.23-apache-2.2.x-linux-ia64.so	2007-07-06 02:20	973K	
mod_jk-1.2.23-apache-2.2.x-linux-ia64.so.asc	2007-07-06 02:20	189	
mod_jk-1.2.23-apache-2.2.x-linux-ppc64.so	2007-07-06 02:20	643K	
mod_jk-1.2.23-apache-2.2.x-linux-ppc64.so.asc	2007-07-06 02:21	189	
mod_jk-1.2.23-apache-2.2.x-linux-x86_64.so	2007-07-06 02:21	743K	
mod_jk-1.2.23-apache-2.2.x-linux-x86_64.so.asc	2007-07-06 02:21	189	

Apache/2.4.1 (Unix) OpenSSL/1.0.0g Server at archive.apache.org Port 80

图 8-14　连接器下载页面

2) 安装和配置 mod_jk

将 mod_jk-1.2.23-apache-2.2.x-linux-i686.so 复制到/etc/httpd/modules 目录中，并将它重命名为 mod_jk.so。在/usr/apache-tomcat-6.0.35/conf 目录中新建子目录 jk，并新建文件 workers.properties，内容如下：

```
workers.tomcat_home=/usr/apache-tomcat-6.0.35/
workers.java_home=/usr/lib/jvm/java
ps=/
worker.list= ajp13
worker.ajp13.port=8009
worker.ajp13.host=192.168.111.133
worker.ajp13.type=ajp13
worker.ajp13.lbfactor=1
```

修改/usr/share/tomcat5/conf/server.xml 文件，在<Engine name="Catalina" defaultHost="localhost">语句下添加以下语句：

```
<Listener className= "org.apache.jk.config. ApacheConfig"
modJk="/etc/httpd/modules/mod_jk.so" />
```

重新启动 Tomcat 服务，这时将自动生成目录/usr/apache-tomcat-6.0.35/conf/auto 和文件/usr/apache-tomcat-6.0.35/conf/auto/mod_jk.conf。将文件/usr/apache-tomcat-6.0.35/conf/auto/mod_jk.conf 复制到/usr/apache-tomcat-6.0.35/conf/jk 目录中，并重命名为 mod_jk.conf-auto。

修改 mod_jk.conf-auto 文件，修改后内容如下。

```
<IfModule !mod_jk.c>
LoadModule jk_module "/etc/httpd/modules/mod_jk.so"
</IfModule>
JkWorkersFile "/usr/apache-tomcat-6.0.35/conf/jk/workers.properties"
JkLogFile "/usr/apache-tomcat-6.0.35/logs/mod_jk.log"
JkLogLevel emerg
<VirtualHost *:80>
ServerName localhost
JkMount /*.jsp ajp13
</VirtualHost>
```

3) 配置 Tomcat

要实现 Apache 和 Tomcat 整合，需要设置 Apache 和 Tomcat 的主目录一致。由于 Tomcat 默认的主目录是/var/lib/tomcat5/webapps/ROOT，因此应编辑 Tomcat 的主配置文件/usr/share/tomcat5/conf/server.xml，找到如下内容：

```
<Host name="localhost" appBase="webapps"
unpackWARs="true" autoDeploy="true"
xmlValidation="false" xmlNamespaceAware="false">
```

在其后添加以下语句：

```
<Context path="" docBase="/var/www/html" debug="0"/>
```

这表明配置 Tomcat 主目录为/var/www/html/。

4) 配置 Apache

编辑文件/etc/httpd/conf/httpd.conf，在文件末尾添加如下内容：

```
Include /usr/apache-tomcat-6.0.35/conf/jk/mod_jk.conf-auto
```

5) 重新启动 Apache 和 Tomcat

由于使用 mod_jk 连接器分别修改了 Apache 和 Tomcat 的配置文件，因此需要使用以下命令重新启动 Apache 和 Tomcat。

```
/etc/init.d/httpd restart
./usr/apache-tomcat-6.0.35/bin/shutdown.sh
./usr/apache-tomcat-6.0.35/bin/startup.sh
```

6) 测试 Apache 和 Tomcat 整合

在主目录/var/www/html/中建立一个名为 test.jsp 的文件，该文件的内容如下：

```
Hello! The time is <%= new java.util.Date() %>
```

在客户端的浏览器中访问“http://Linux 服务器的 IP 地址/test.jsp”，如果出现如图 8-15 所示的“Hello! The time is 当前时间”的信息，则表示 Apache 和 Tomcat 整合成功。

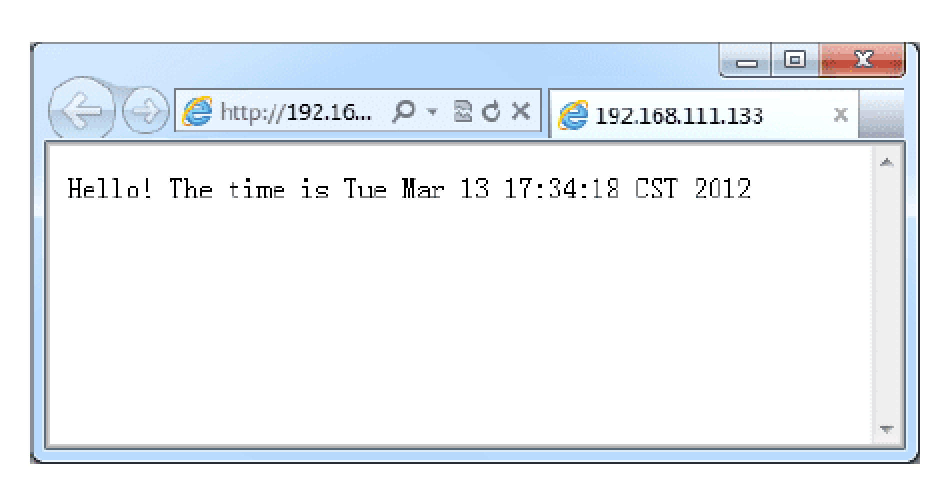


图 8-15　整合测试结果

8.9　本 章 小 结

本章主要介绍了使用 Apache 服务器软件架设 Web 服务器的方法。先后讲述了有关 HTTP 协议的相关知识，Apache 服务器的安装、运行与配置，最后还介绍了 Apache 服务器对动态网页技术的支持方法。作为 Linux 的 Web 服务中最流行的技术之一，读者应该熟练掌握 Apache 服务器各种功能的配置方法。

8.10 课 后 习 题

1. 填空题

(1) URI 是__________的缩写。

(2) Web 服务通常可以分为两种：____________和__________。

2. 选择题

(1) apache 是（ ）。

A. 一种 WEB Server　　B. 一种 FTP Server

C. 一种 News Server　　D. 一种 Mail Server

(2) CGI 是（ ）。

A. 一种后端动态程序　　B. 一种 ASP 程序

C. 一种 PHP 程序　　D. 一种后端程序接口

3. 简答题

(1) 如何启动、终止、重新启动和查看 WWW 服务?

(2) 虚拟主机有哪两种？主机数有什么限制?

4. 操作题

(1) 在 Web 服务器中建立一个名为 temp 的虚拟目录，其对应的物理路径是/usr/local/temp，并配置 Web 服务器允许该虚拟目录具备目录浏览和允许内容协商的多重视图特性。

(2) 使用 192.168.1.17 和 192.168.1.18 两个 IP 地址创建基于 IP 地址的虚拟主机，其中 IP 地址为 192.168.1.17 的虚拟主机对应的主目录为/usr/www/web1，IP 地址为 192.168.1.18 的虚拟主机对应的主目录为/usr/www/web2。

(3) 让 Web 服务器支持 CGI 运行环境。

(4) 让 Web 服务器支持 PHP 运行环境。

(5) 让 Web 服务器支持 JSP 运行环境。

第 9 章

FTP 服务的配置及应用

FTP 是互联网中非常常见的一种文件传输方式，通过它可以实现网络中稳定高速的文件传输。在 Linux 的各个发行版本中，使用的 FTP 服务器软件有很多，例如 Wu-ftpd、Proftpd 以及 VsFTPd 等。而在目前 Red Hat Linux 的各个发行版本中，都是以 VsFTPd 为默认的 FTP 软件，所以在本章中，我们将以 VsFTPd 为例介绍 FTP 服务的配置和应用。

9.1 FTP 服务概述

FTP(File Transfer Protocol，文件传输协议)是基于 TCP/IP 协议的服务，用户在网络中的文件传输与其他协议(HTTP、SFTP、TFTP 等)相比，FTP 协议传输文件更加稳定、迅速。至今，FTP 服务仍然是网络中使用广泛的服务之一。

9.1.1 FTP 的工作原理

FTP 的工作原理如图 9-1 所示。FTP 服务器首先监听 TCP 的连接，等待客户端的下载或者上传请求。其控制连接和数据连接均为 TCP 连接，控制连接主要用于传送用户名、密码以及控制传输方式等信息，而数据连接用于传送文件数据。客户端和服务器都运行着控制进程和数据传输进程。

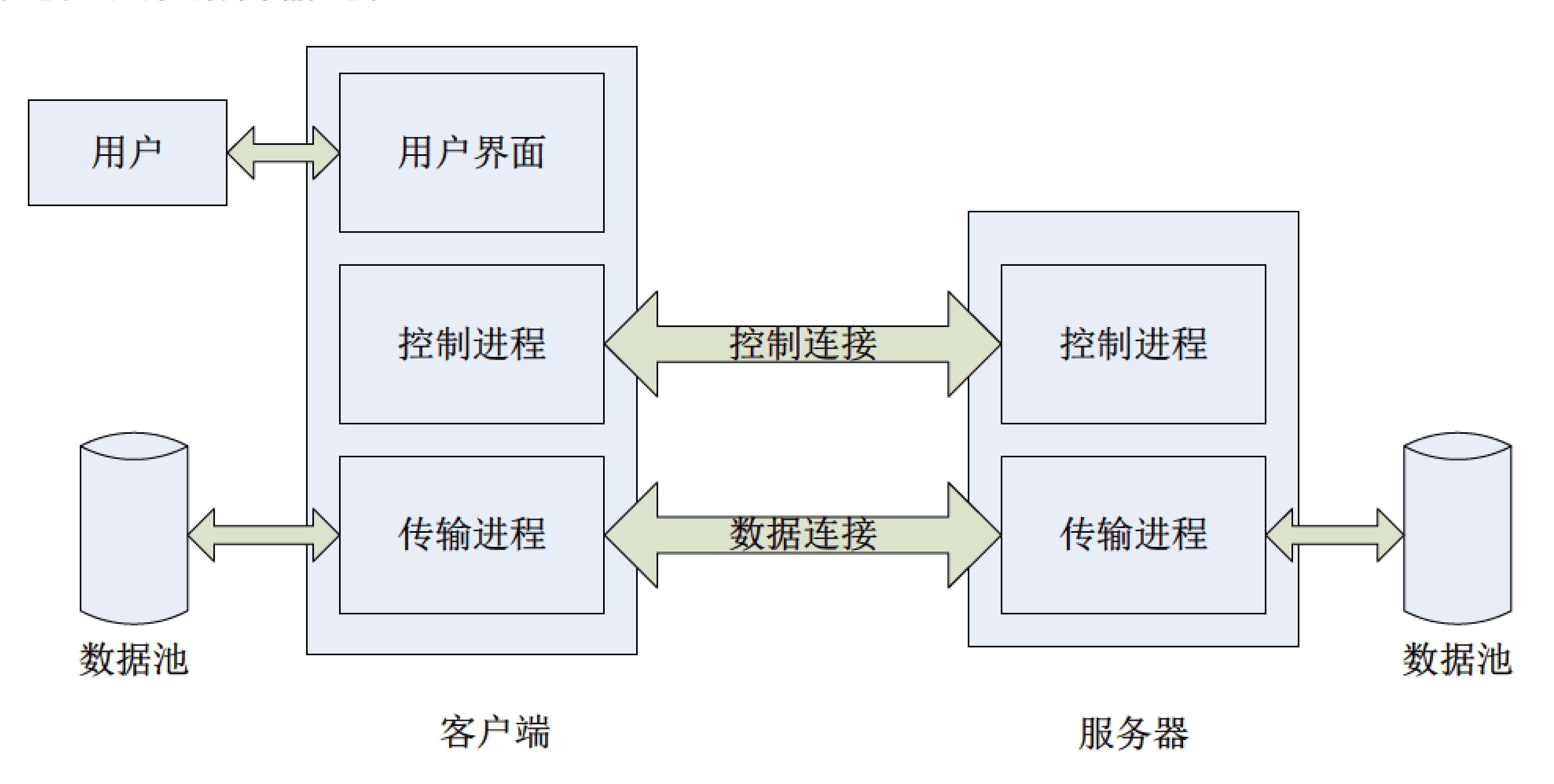


图 9-1 FTP 工作原理

当用户需要从服务器下载文件时，客户端的控制进程会发起一个 TCP 连接请求，服务器端监听进程接受到请求后，建立控制连接，此时，双方就可以传输控制信息了。为了传输数据，此时双方还需要再建立一个数据连接。

控制连接默认情况下，使用的是服务器端的 TCP 端口号 21，同时，客户端还会在控制信息中附带自己的一个空闲端口，服务器会再使用 TCP 端口号 20 与客户端所提供的空闲端口建立数据连接，然后开始传输数据。

提示： FTP 协议之所以需要建立两个连接，是因为在传输数据的过程中，双方仍然需要传输控制数据，为了让两种数据互不干扰，就需要建立两个 TCP 连接分别传输两种数据。

在客户端与服务器传输数据的过程汇总，控制连接是一直存在的，但数据连接在传输完一个文件后就会释放连接，如果要传输另一个文件，则需要重新创建一个数据连接。这一特性决定了 FTP 在传输大量的小文件时效果比较低，每传输一个文件都需要创建和释

放数据连接，而不像有些协议(例如 Samba)，可以在一个连接中将所有的文件一次传输完毕。

另外，FTP 的工作模式和其他网络通信协议也有很大的不同，如使用 HTTP 协议传输数据时，只使用一个连接进行通信。

9.1.2　FTP 的连接模式

FTP 的连接模式分为 POST 模式(Standard，也称为主动模式)和 RASV 模式(Passive，也称被动模式)两种。在 FTP 服务器上同时支持这两种连接模式，但具体使用哪种模式与 FTP 客户端有关。

1. POST 模式

FTP 客户端以 POST 模式连接 FTP 服务器时，首先使用一个随机选择的端口(1024 端口以下)和 FTP 服务器的 21 端口建立控制连接，当客户端需要接收数据时，在控制连接发送 POST 命令。此命令还包含了客户端接收数据的端口号，在传输数据时，服务器通过 TCP20 端口与客户端指定的端口发送数据。

2. RASV 模式

FTP 客户端使用 RASV 模式连接服务器时，建立控制连接的过程与 POST 模式类似，但建立连接时使用的是 RASV 命令，而不是 POST 命令。FTP 服务器接收到 RASV 命令后，随机打开一个高数字的端口(大于 1024)。并且通知客户端在这个端口上传输数据的请求，客户端连接 FTP 服务器该端口，然后 FTP 服务器将通过这个端口进行数据的传输。

提示： 很多服务器的防火墙为了安全考虑，不允许接收外部发起的连接，客户端也就无法通过防火墙打开服务器的连接端口，所以很多处于防火墙后的 FTP 服务器不支持 RASV 模式；而许多内网用户的客户端因为服务器的 TCP 连接无法与内部网络的客户端建立一个新的连接，也就不能使用 POST 模式登录 FTP 服务器。所以在创建 FTP 服务器时要对防火墙进行相应的配置，以保证 FTP 服务器的正常工作。

9.1.3　数据传输模式

FTP 在工作过程中使用了专用的数据连接，在传输文件数据时，FTP 协议规范提供了控制未见传送与存储的多种选择，包括文件类型、格式控制、文件结构和传输方式 4 个方面中所规定的选项中确定一种。

1. 文件类型

FTP 可选择的文件类型有 4 种：ASCII 码、EBCDIC 码、二进制文件以及本地文件类型。

1) ASCII 码

ASCII 码文件类型也成为文本类型，是 FTP 传输的默认选项。该选项使文件数据以

ASCII 码的形式在数据连接中传输。在传输数据前，发送方先将本地文件转换成 ASCII 码的形式，再发送到网络，而接收方则将从网络中接收到的 ASCII 码还原为本地文件格式，再写入硬盘中。

2) EBCDIC 码

EBCDIC 码也成为广义二进制码或者十进制交换码。是一种字母或数字字符的二进制编码，每个字母或者数字都被表示为一个 8 位的二进制数。当采用该选项传输时，要求两端都是 EBCDIC 系统。

3) 二进制文件类型

二进制文件类型也称为图像文件类型，传输时是连续的比特流，没有任何格式，通常用于传输二进制文件。

4) 本地文件类型

当服务器与客户端系统所规定的字节位数不同时，需要使用该选项。

2. 格式控制

格式控制只作用于 ASCII 码和 EBCDIC 码两种文件类型，具有 3 种选项：非打印选项、远程登录格式控制选项和 Fortran 回车控制选项。

1) 非打印选项

是默认选项，表示此文件中不包含有垂直格式信息。

2) 远程登录格式控制选项

表示文件中含有向打印机解释的远程登录垂直格式控制符。

3) Fortran 回车控制选项

表示每行首字符都是 Fortran 格式控制符。

3. 数据结构

数据结构也有 3 个选项：文件结构、记录结构和页结构。

1) 文件结构

为默认选项，认为数据是一个连续的字节流，不存在其他结构。

2) 记录结构

该选项只用于文本文件，认为数据是由一条记录组成的。

3) 页结构

页结构在发送数据时规定包含页号，以便接收方能随机地存储各页。

4. 传输方式

传输方式一共有 3 种选项：流方式、块方式和压缩方式。

1) 流方式

流方式是默认的方式，此方式规定文件将以字节流的形式传输。对于文件结构，发送方在文件结束处提示关闭数据连接。对于使用记录结构的传输，有专用的两个字节序列码标志记录结束和文件结束。

2) 块方式

文件将作为一些列的块来传输，每一个块的前面都带有一个或者多个首部字节。

3) 压缩方式

该方式用一个简单的全长编码压缩方式，压缩连续出现的相同字节，再发送。由于发送方可以使用更好的方法实现压缩文件，因此此方式已经很少使用。

对于以上 4 个方面的各个选项，FTP 的连接双方必须在数据传输前都事先确定。虽然 FTP 提供了如此丰富的选择方式，但由于操作系统和软件的局限性，很多方式现在已经废弃不用了，为了系统兼容性和稳定性的考虑，当今主流的 Windows 和 Linux 操作系统平台的 FTP 客户端和服务器对上述的选项进行如下的限制：

- 文件类型：只允许 ASCII 码或者二进制文件类型。
- 格式控制：只允许非打印选项。
- 数据结构：只允许文件结构。
- 传输方式：只允许流方式。

也就是说，在实际的使用中，只会涉及 ASCII 码和二进制文件类型两种选择，其他的选项都已经使用了默认选项，不用修改。

9.1.4　FTP 的控制命令

当客户端与服务器建立控制连接后，客户端的控制进程就可以通过该连接向服务器发送控制指令了。服务器端的监听进程随时都可以接收客户端发送的请求，然后根据指令的内容作出相应的工作，再将结果反馈给客户端。

控制指令以 ASCII 码字符的形式传输，每个指令由 3～5 个大写的 ASCII 字符组成，一些指令后面还可以附带参数，指令和参数之间以空格分开，并以一对回车和换行符(CR/LF)作为指令的结束标志，FTP 中常见的控制命令如表 9-1 所示。

表 9-1　FTP 常用指令及其含义

指　令	参　数	含　义
ABOR	无	要求服务器终止一次 FTP 服务命令及所有相关的数据传输
ALLO	N	要求服务器保留 n 个字节的存储空间用于存放将要传输的文件
APPE	文件名	要求服务器准备接收一个文件，如果同样的文件在服务中已存在，则追加到此文件之后
CDUP	无	返回服务器当前目录的上级目录
CWD	路径	把服务器上指定的路径变为当前目录
DELE	文件名	删除服务器上的指定文件
HELP	指令名	返回指定指令的帮助信息，如果未指定指令名，则返回所有指令的帮助信息
LIST	路径名	要求服务器返回其指定路径下的所有文件及目录，如果没有指定路径，则返回当前目录下的所有文件和目录
MKD	路径名	要求服务器在指定路径上创建目录
MODE	S、B、C	设置服务器传输方式，S 为流方式，B 为块方式，C 为压缩方式
NLIST		要求服务器返回其指定路径下的所有目录，如果没有指定路径，则返回当前目录下的所有目录

续表

指 令	参 数	含 义
NOOP	无	空操作，有些 FTP 服务器设置了空闲断开的功能，如果客户端长时间不发送任何数据，服务器将主动断开控制和传输连接，发送指令可以维持连接
PASS	密码	向服务器发送要求登录的用户名和密码
PASV	无	告诉服务器在一个非标准接口上监听客户端的数据连接
POST	6 个数字	为数据连接指定一个客户端的 IP 地址和端口
PWD	无	返回当前工作目录的名称
QUIT	无	释放控制连接
REST	n	指定一个文件起始位置的偏移值，从此偏移值开始传输文件
RETR	文件名	从服务器复制一个指定的文件到客户端
RMD	路径名	在服务器上删除指定目录
RNFR	文件名	指定要重命名的文件，后面应该紧跟 RNTO 命令
RNTO	文件名	把 RNFR 指定的文件改为该文件名
STAT	目录名	要求服务器以应答形式发送状态
STOR	文件名	要求服务器接收指定的文件，如果服务器上有同名的文件，则直接覆盖
STOU	文件名	要求服务器接收指定的文件，如果服务器上有同名的文件，则报错
SYST	无	要求服务器发送其操作系统类型
TYPE	A、E、I	确定数据传输方式。A 为 ASCII 码方式，E 为 EBCDIC 方式、I 为二进制方式
USER	用户名	指定登录服务器的用户名

提示： 此处我们介绍的控制指令属于协议级的指令，与稍后介绍的用户使用 FTP 客户端时输入的命令并不一样，但很多用户命令确实是由控制指令而来的。

9.1.5 FTP 的匿名访问

从操作的安全性来考虑，所有对服务器的访问都应该经过授权后才能进行，很多服务和协议也是这么做的(例如，数据库访问、邮件服务)。FTP 服务同样具有这样的功能，在 FTP 客户端与服务器建立控制连接后，要先提供 USER 和 PASS 指令才能登录，然后服务器才会接受其他命令。登录 FTP 服务器的命令格式为：

```
ftp://用户名:密码@服务器名或者 IP 地址
```

但是，为了网络共享的考虑，FTP 服务的提供者可能希望对互联网的所有用户都提供某些文档或者程序的下载功能，也就是说，FTP 的内容希望是公开的，此时，每个用户都必须获得用户账号才能登录 FTP 服务器就显得有些多余，而且 FTP 协议已经提供了访问权限的控制功能，可以很好地对用户的访问进行规范。综合上面的条件，FTP 协议又规定了一种匿名账号的机制，用户可以使用一个通用的账号来登录系统，然后就可以发送 FTP 指令对服务器进行操作。当热，为了安全考虑，这个匿名账号的权限一般都是有限的，一般只能做到列出目录、下载文件等读取的操作。当然，并不是所有的 FTP 服务器都必须支持匿名账号，这些功能都是由服务器管理员根据需要配置的。

根据 FTP 协议的规定，匿名账号的用户名统一为 anonymous，密码要求是一个 Email 地址，但在大部分情况下，密码可以是任意字符串。

可以说，匿名账号的引入大大方便了用户访问 FTP 服务器，这是 FTP 服务能够在互联网上使用如此广泛的重要原因之一。

9.2　VsFTPd 的安装与运行

VsFTPd(Very Secure FTP Daemon)是一种遵循 GPL 协议的开源 FTP 服务器软件，具有安全、快速、稳定的特点，可以在 UNIX 和 Linux 系统下运行。下面我们先介绍它的主要特性，然后介绍安装与运行。

9.2.1　VsFTPd 的主要特性

VsFTPd 作为一个高度安全的 FTP 服务器软件，加之其完善的功能和突出的性能，主要具有以下特点：

- 运行稳定。VsFTPd 可以在单机上支持 4000 个以上的并发用户同时连接。
- 支持基于 IP 的虚拟 FTP 服务器。
- 支持虚拟用户。
- 支持 PAM 或者 xinetd_tcp_wrappers 的认证方式。
- 支持两种运行方式：PAP 与 xinetd。
- 支持每个虚拟用户具有独立的配置。
- 支持带宽限制。

9.2.2　VsFTPd 的安装

VsFTPd 的安装有两种方式：一种是通过 RPM 软件包的方式，这种安装方法比较简单，推荐使用这种安装方法；另一种是通过编译源代码进行安装。

1. 使用 RPM 软件包安装

首先我们要将 RPM 软件包从操作系统的安装光盘复制到硬盘中，在 CentOS 5.5 的安装光盘中，VsFTPd 软件包的位置在/CnetOS/vsftpd-2.0.5-16.el5_4.1.i386.rpm，复制文件后，运行以下的命令进行安装。

```
[root@CentOS ~]# rpm -ivh vsftpd-2.0.5-16.el5_4.1.i386.rpm
```

安装成功后，添加的重要文件及位置如下：

- /usr/sbin/vsftpd：vsFTPd 服务器的进程文件。
- /etc/vsftpd vsftpd.conf：vsFTPd 服务器的配置文件。
- /etc/pam.d/vsftpd：vsFTPd 服务器认证本地用户的 PAM 接口配置文件。
- /usr/share/doc/vsftpd-2.0.5：帮助和说明文档存放的目录。
- /usr/share/man：vsFTPd 帮助文件安装位置。
- /var/ftp：vsFTPd 服务器匿名用户的工作目录。

2. 使用源代码安装方式

VsFTPd 的源代码可以从网站获取，下载的网址为 ftp://vsftpd.beasts.org/users/cevans/，我们需要下载打包文件，其后缀为“.tar.gz”，然后将文件下载到当前目录，下面我们以 vsftpd-2.3.1.tar.gz 为例介绍源代码的安装过程。

```
[root@CentOS ~]# rpm -q vsftpd                      #查看是否已经安装 vsFTPd
vsftpd-2.0.5-16.el5_4.1
[root@CentOS ~]# rpm -e vsftpd-2.0.5-16.el5_4.1     #先卸载已经安装的 vsFTPd
[root@CentOS ~]# tar -xvzf vsftpd-2.3.1.tar.gz      #解压源码包
…
[root@CentOS ~]# cd vsftpd-2.3.1
[root@CentOS vsftpd-2.3.1]# make                    #编辑源码
[root@CentOS vsftpd-2.3.1]# make install            #开始安装
#安装完成后，用户还需要手动将配置文件复制到指定位置
[root@CentOS ~]# cp ./vsftpd.conf /etc              #复制配置文件到/etc 目录
[root@CentOS ~]# mkdir /var/ftp                     #手动创建 ftp 匿名访问目录
[root@CentOS ~]# mkdir /var/ftp/pub
```

9.2.3 VsFTPd 的运行

安装完成后，我们可以输入以下的命令来启动 vsFTPd 进程。

```
[root@CentOS /]# /usr/sbin/vsftpd
```

此条命令可以放到/etc/rc.local 文件中作为开机启动选项。
启动 vsFTPd 进程后，我们可以使用 ps 命令来查看进程是否已经正常运行，如下所示：

```
[root@CentOS /]# ps -eaf | grep vsftpd
root      8858     1  0 10:39 ?        00:00:00 /usr/sbin/vsftpd
root      8873  4605  0 10:39 pts/0    00:00:00 grep vsftpd
```

再使用 netstat 命令来查看服务器的 21 端口是否已经处于监听状态。

```
[root@CentOS /]# netstat -tnl | grep :21
tcp        0      0 0.0.0.0:21          0.0.0.0:*                   LISTEN
```

最后，我们可以在本机上对 FTP 进行简单的测试，查看其功能是否正常，如下所示：

```
[root@CentOS /]# ftp 127.0.0.1
Connected to 127.0.0.1.
220 (vsFTPd 2.0.5)
530 Please login with USER and PASS.
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
Name (127.0.0.1:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (127,0,0,1,114,1)
150 Here comes the directory listing.
drwxr-xr-x    2 0        0            4096 Dec 16  2009 pub
226 Directory send OK.
```

```
ftp> bye
221 Goodbye.
```

从上面的过程我们可以看出，VsFTPd 已经能够正常的登录和退出了。

9.3　VsFTPd 服务器的配置

VsFTPd 的服务器配置主要包括 vsftpd.conf 的配置、匿名用户的配置、虚拟主机的配置和虚拟用户的配置。

9.3.1　vsftpd.conf 的配置

VsFTPd 服务的主要功能都是通过配置文件 vsftpd.conf 来设置的，配置文件中以“选项名=值”的形式对某一选项进行设置，如果选项中没有进行设置，那么服务器将采用默认值。

```
# Example config file /etc/vsftpd/vsftpd.conf
anonymous_enable=YES                #允许匿名用户登录 FTP
local_enable=YES                    #允许本地用户登录 FTP
write_enable=YES                    #允许写入操作的 FTP 控制命令，如 STOR、DELE 等。
local_umask=022                     #设定文件初始权限值
#anon_upload_enable=YES
#anon_mkdir_write_enable=YES
dirmessage_enable=YES                #发送欢迎信息
xferlog_enable=YES                   #开启日志
connect_from_port_20=YES             #设定 20 端口为数据连接端口
#chown_uploads=YES
#chown_username=whoever
#xferlog_file=/var/log/xferlog
xferlog_std_format=YES                #设定日志的格式
#idle_session_timeout=600
#data_connection_timeout=120
#nopriv_user=ftpsecure
#async_abor_enable=YES
#ascii_upload_enable=YES
#ascii_download_enable=YES
#ftpd_banner=Welcome to blah FTP service.
#deny_email_enable=YES
#banned_email_file=/etc/vsftpd/banned_emails
#chroot_list_enable=YES
#chroot_list_file=/etc/vsftpd/chroot_list
#ls_recurse_enable=YES
listen=YES                            #开启端口监听
#listen_ipv6=YES
pam_service_name=vsftpd                 #设定 pam 服务配置文件名
userlist_enable=YES                     #开启用户列表
tcp_wrappers=YES                       #连接请求由 tcp_wrappers 完成访问控制
```

下面我们对这些设置的功能进行详细的介绍。

1) anonymous_enable=YES

此项为开启 FTP 服务器的匿名访问功能，虽然匿名访问使 FTP 更容易传播和推广，

但也带来很大的安全隐患，如果开启此项，那么需要在防火墙配置、服务器配置和 FTP 配置等多方面采取措施来保证服务器的安全。对于只提供给特定用户、或者是私有的 FTP，一般不开启此服务，将其值设置为“NO”。

2) local_enable=YES

此项为开启本地账号登录 FTP 服务器，本地账号包括操作系统账号和虚拟账号(稍后介绍)。但是，开启此项也并不意味着本地用户肯定能够登录，这还取决于 PAM 和虚拟账号等的正确配置。

3) write_enable=YES

此项允许服务器接收与写有关的控制指令，包括 STOR、DELE、RNFR、RNTO、MKD、RMD、APPE、SITE 指令。

4) local_umask=022

设置本地用户创建新的文件时的默认权限值，实际上，“022”是一个八进制数，表示初始的权限值是创建者的全部权限，而其他用户只有读取和执行的权限。另外，local_umask 的值还可以设为“077”，表示创建者拥有全部权限，其他用户没有权限。

5) anon_upload_enable=YES 和 anon_mkdir_write_enable=YES

这两个选项是开启服务器允许匿名用户上传文件和创建目录的权限。如果要真正允许用户写入，还需要将 write_enable 选项也设置为“YES”。但是开启这两个选项会给服务器的安全带来很大的风险，一般都设置为“NO”，默认值也是“NO”。

6) dirmessage_enable=YES

此项表示用户第一次登录新目录时，会发送给用户一些提示信息，这些信息默认存放在该目录的.message 文件中，但可以通过 message_file 选项进行更改。

7) xferlog_enable=YES、xferlog_file=/var/log/xferlog 和 xferlog_std_format=YES

这是一组相关的配置，用来启用 VsFTPd 的日志功能，将日志路径及文件名设置为/var/log/xferlog，采用与其他 FTP 服务器兼容的格式。VsFTPd 的日志详细记录了用户的登录、上传、下载和退出等操作信息，日志格式由 xferlog_std_format 选项决定，默认值为“NO”，将使用 VsFTPd 独有的可读性更好的格式。

8) connect_from_port_20=YES

此选项规定 FTP 服务器采用主动模式与客户端建立连接的时候是否将端口固定为 20，这主要是为了配合客户端的设置。此项的默认值为“NO”，表示 VsFTPd 可以使用 1024 以上的端口来建立连接。

9) chown_uploads=YES 和#chown_username=whoever

这是一组先关的选项，表示所有匿名用户上传的文件其所有者将都设置为 whoever，这样设置主要是为了安全考虑，即匿名用户默认只能访问使用匿名账号上传的文件。

10) idle_session_timeout=600

此选项设置控制连接超时的时间，单位为秒，当客户端不发送任何交互指令超过 600 秒时，服务器将主动断开与其的控制连接，这样主要是为了减轻服务器的负担，释放更多的空间资源给正在使用的客户端。不过，在前面的“FTP 控制指令”小节中我们介绍过，客户端也可以通过发送空操作“NOOP”来维持控制连接。

11) data_connection_timeout=120

此项设置数据连接的超时时间，单位为妙，如果客户端在建立连接后 120 秒没有请求或者上传任何数据，那么服务器将主动断开数据连接。

12) nopriv_user=ftpsecure

此选项给出了当 VsFTPd 处于非特权模式运行时，所使用的用户身份，此选项的默认值为“nobody”。处于安全考虑，管理员可以将 VsFTPd 设置为非特权模式运行，此时 VsFTPd 进程就需要一个用户身份来运行，但“nobody”用户被很多其他 Linux 软件使用，所以需要管理员单独为 VsFTPd 创建一个用户，此处为“ftpsecure”。

13) async_abor_enable=YES

此选项设置 FTP 服务器是否接受 async ABOR 指令。开启此选项会给服务器带来安全风险，所以默认值为“NO”，但有些 FTP 客户端会需要支持此指令，否则客户端将无法正常使用。

14) ascii_upload_enable=YES 和 ascii_download_enable=YES

这两个选项设置服务器是否允许 ASCII 码模式，当 FTP 服务器使用 ASCII 传输模式时，容易受到 DOS 攻击，为了避免攻击，可以将这两个选项设置为“NO”，VsFTPd 会发送给客户端允许 ASCII 模式，但实际上使用的是二进制模式。

15) ftpd_banner=Welcome to blah FTP service

此选项设置当用户登录时，显示的欢迎信息。此选项的默认值是显示 VsFTPd 服务器的名称和版本信息，这给黑客和攻击者提供了有价值的信息，所以一般情况下我们需要使用该选项，将欢迎信息设置为其他内容，如果希望显示的文本内容较多，还可以将所有需要显示的内容放入一个文件中，再使用 banner_file 选项来指定这个文件。

16) chroot_list_enable=YES 和 chroot_list_file=/etc/vsftpd/chroot_list

这两个选项用来打开并指定用户列表功能。当 chroot_local_user 选项设为“NO”时，这些用户登录 FTP 后所看到的根目录即为自己的个人目录，并且也不能切换到其他目录，这样做可以保证服务器的安全，用户无法查看服务器的主机文件系统。当 chroot_local_user 设置为“YES”时，上述两个选项设置的用户列表中的用户不会被限制在个人目录中，可以查看服务器中的其他目录，而不在这个列表中的用户将会被限制在个人目录中。

17) ls_recurse_enable=YES

此选项设置客户端在发送 ls 命令时是否可以附带“-R”选项。因为“-R”选项会列出整个目录及其子目录中的所有文件，当文件和目录层次较多时将耗费大量的服务器资源，如果恶意用户利用这个漏洞，将造成 DOS 攻击，因此，VsFTPd 此项的默认值是“NO”。

18) listen=YES 和 listen_ipv6=YES

这两个选项使 VsFTPd 将以独立的方式运行，前者监听 IPv4 端口，后者用来监听 IPv6 端口，但两者不能在同一个配置文件中设置。除了独立运行方式外，VsFTPd 还可以使用 inetd 方式，它对网络的安全有更多的控制。

19) anon_max_rate、local_max_rate、max_clients 和 max_per_ip

这 4 个选项用来对用户的速率和连接数做出限制。虽然没有在默认的 vsftpd.conf 配置

中给出，但是在实际使用中绝大多数管理员都会使用，因为服务器的资源总是有限的，如果不对客户的行为做出限制，很容易会造成网络或者主机的瘫痪。

anon_max_rate 选项用来设置匿名用户访问的最大速率，单位为 b/s，如果设置为 0，则表示无限制。

local_max_rate 选项是设置本地用户访问的最大速率，和 anon_max_rate 选项的使用方法类似。

max_clients 选项用来限制 VsFTPd 的最大客户端连接数，如果设置为 0，则表示无限制。

max_per_ip 选项用来限制每个 IP 的最大连接数，如果设置为 0，则表示无限制。

9.3.2 匿名用户的配置

在某些情况下，FTP 服务器管理员为了方便用户的使用，会开通匿名账号，但是支持匿名账号又会带来很多安全问题，如果配置不当，很容易造成主机信息的泄露，或者遭到黑客的攻击。因此，VsFTPd 提供了很多有关匿名账号的设置功能用来管理匿名账号。下面，我们通过一个简单的实例来讲述如何配置匿名账户。

在配置匿名账号的过程中，我们要达到以下的目的：

- 支持匿名账户访问。
- 匿名账号登录 FTP 后进入/var/ftp/pub 目录，并且将此目录限定为匿名账号的根目录。
- 匿名用户可以将文件上传到/var/ftp/pub/upload 目录，但不能下载或者删除该目录中的内容。
- 匿名账号使用 aaa@(这是匿名账号的默认登录密码)作为密码登录时将被拒绝，这主要是为了防止恶意的登录攻击。

设置匿名账号的具体操作步骤如下：

1) 设置匿名账号权限

打开 vsftpd.conf 文件，设置或者添加以下选项：

```
anonymous_enable=YES              #允许匿名用户登录 FTP
anon_world_readable_only=NO        #开放匿名账号的写权限
anon_root=/var/ftp/pub             #设置匿名账号根目录
anon_upload_enable=YES            #开启上传权限
chown_uploads=YES                 #上传文件所有者为 root，而不是匿名账户
```

2) 创建上传目录

```
[root@CentOS /]# mkdir /var/ftp/pub/upload
[root@CentOS /]# ftp /var/ftp/pub/upload
```

3) 禁止默认登录密码

在 vsftpd.conf 中加入以下的选项。

```
deny_email_enable=YES                              #开启阻止邮箱登录功能
banned_email_file=/etc/vsftpd/banned_email        #设置密码登录阻止列表文件位置
```

然后在/etc/vsftpd 目录中创建 banned_email 文件，并输入以下的内容。

```
aaa@
```

4) 重启 FTP 服务

使用 kill 命令重启 VsFTPd 进程，使设置生效。

```
[root@CentOS /]# kill -HUP vsftpd
```

以上的步骤完成后，我们可以登录 FTP 进行测试。测试内容如下：

```
C:\Users\Administrator>ftp 192.168.111.133        #连接 FTP
连接到 192.168.111.133。
220 (vsFTPd 2.0.5)
用户(192.168.111.133:(none)): anonymous          #匿名登录
331 Please specify the password.
密码:                                             #输入 aaa@
530 Login incorrect.
登录失败。
ftp> user anonymous                               #再次匿名登录
331 Please specify the password.
密码:                                             #输入任意字符
230 Login successful.
ftp> ls -l                                        #查看根目录内容
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-------    1 0        0              0 Feb 21 02:14 ftpfile1
-rw-------    1 0        0              0 Feb 21 02:14 ftpfile2
drwxr-xr-x    2 0        0           4096 Feb 21 02:24 upload226 Directory
send OK.
226 Directory send OK.
ftp: 收到 132 字节，用时 0.00 秒 132000.00 千字节/秒。
ftp> pwd                                          #查看目录位置
257 "/"
ftp> get ftpfile1                                 #试图下载文件 ftpfile1
200 PORT command successful. Consider using PASV.
550 Failed to open file.
ftp> cd upload                                    #进入 upload 目录
250 Directory successfully changed.
ftp> ls -l
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
ftp> put upfile.txt                               #上传文件
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 File receive OK.                              #上传成功
ftp: 发送 118 字节，用时 0.00 秒 151000.00 千字节/秒。
ftp> ls -l                                        #查看上传文件权限
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-------    1 0        0             18 Feb 21 02:32 upfile.txt
226 Directory send OK.
ftp: 收到 68 字节，用时 0.00 秒 68000.00 千字节/秒。
ftp> get main.c                                   #试图下载上传的文件
200 PORT command successful. Consider using PASV.
550 Failed to open file.                          #下载失败
ftp> rm main.c                                    #试图删除上传文件
```

```
550 Permission denied.                        #删除失败
ftp>
```

9.3.3 虚拟主机的配置

在 VsFTPd 中，虚拟主机是指在一台服务器中配置多个 VsFTPd 服务，每个 VsFTPd 服务都可以采用不同的配置，以适应不同类型的用户，而且从客户端来看，这些 FTP 服务器是在不同主机上运行的。VsFTPd 虚拟主机的原理是将每个 VsFTPd 服务绑定在不同的 IP 地址上。

VsFTPd 的一个典型应用是基于多网卡的服务器，例如，一个服务器同时拥有一块网卡连接外网，另一块网卡连接内网，此时，就可以在两块网卡上运行不同配置的 VsFTPd 服务。当内网用户访问 VsFTPd 服务时可以获得较多的权限；而当外网用户访问 VsFTPd 服务时，将受到很多限制，而这两个 FTP 服务又可以共享部分文件系统。

除了多网卡的应用，VsFTPd 虚拟主机也可以在一块网卡的服务器中使用，此时管理员需要在网卡上配置子接口，得到一块逻辑网卡，而不同的逻辑网卡可以设置不同的 IP 地址。这样，我们也可以达到在一个网卡上运行多个 VsFTPd 服务的目的。

下面，我们将以添加逻辑网卡的方式讲述创建虚拟主机的过程。

(1) 创建虚拟网卡。

使用 root 用户登录 Linux，查看当前网卡的 IP 设置情况。并创建一个新的虚拟网卡，设置的命令如下：

```
[root@CentOS /]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:8B:14:35
          inet addr:192.168.111.133  Bcast:192.168.111.255
Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe8b:1435/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:988 errors:0 dropped:0 overruns:0 frame:0
          TX packets:597 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:722837 (705.8 KiB)  TX bytes:50302 (49.1 KiB)
          Interrupt:67 Base address:0x2024
…
[root@CentOS /]# ifconfig
[root@CentOS /]# ifconfig eth0:1 192.168.111.134 netmask 255.255.255.0 up
[root@CentOS /]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:8B:14:35
          inet addr:192.168.111.133  Bcast:192.168.111.255
Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe8b:1435/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:988 errors:0 dropped:0 overruns:0 frame:0
          TX packets:605 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:722837 (705.8 KiB)  TX bytes:51324 (50.1 KiB)
          Interrupt:67 Base address:0x2024

eth0:1    Link encap:Ethernet  HWaddr 00:0C:29:8B:14:35
          inet addr:192.168.111.134  Bcast:192.168.111.255
Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```
        Interrupt:67 Base address:0x2024
...
```

(2) 停止 VsFTPd 服务。

运行 killall 命令暂停 VsFTPd 服务。

```
[root@CentOS /]# killall vsftpd
```

(3) 在 VsFTPd.conf 中增加配置：

```
listen_address=192.168.111.133
```

(4) 重启 VsFTPd 服务：

```
[root@CentOS vsftpd]# /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf &
```

(5) 为第二个 VsFTPd 服务器建立匿名账号对应的本地账号和个人目录：

```
[root@CentOS /]# useradd -d /var/ftp2 -s /sbin/nologin ftp2
```

(6) 改变匿名账号目录权限，使其没有写权限：

```
[root@CentOS /]# chown root /var/ftp2
```

(7) 复制一份配置文件 vsftpd.conf 并重新命名：

```
[root@CentOS /]# cp /etc/vsftpd/vsftpd.conf /etc/vsftpd/vsftp2d.conf
[root@CentOS /]# cd /etc/vsftpd/
[root@CentOS vsftpd]# ls
banned_email  user_list     vsftpd.conf
ftpusers     vsftp2d.conf  vsftpd_conf_migrate.sh
```

(8) 在新的配置文件 vsftp2d.conf 中修改监听 IP 地址，并添加新的配置选项：

```
ftpd_banner=this is ftp2.
ftp_username=ftp2
listen=YES
listen_address=192.168.111.134
```

(9) 使用 vsftp2d.conf 启动新的 VsFTPd 进程：

```
[root@CentOS /]# /usr/sbin/vsftpd /etc/vsftpd/vsftp2d.conf
```

经过以上的步骤，新的 VsFTPd 服务已经创建完毕，下面我们再对其进行测试。

```
C:\Users\Administrator>ftp 192.168.111.133    #先登录原 FTP
连接到 192.168.111.133。
220 (vsFTPd 2.0.5)                            #欢迎信息为默认信息
用户(192.168.111.133:(none)): anonymous
331 Please specify the password.
密码:
230 Login successful.
ftp> ls -l                                    #查看根目录内容
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-------    1 0        0               0 Feb 21 02:14 ftpfile1
-rw-------    1 0        0               0 Feb 21 02:14 ftpfile2
drwxr-xr-x    2 0        0            4096 Feb 21 02:33 upload
226 Directory send OK.
ftp: 收到 196 字节，用时 0.00 秒 196.00 千字节/秒。
ftp> bye
```

```
221 Goodbye.

C:\Users\Administrator>ftp 192.168.111.134      #登录新的 FTP
连接到 192.168.111.134。
220 this is ftp2.                               #欢迎信息已经修改
用户(192.168.111.134:(none)): anonymous
331 Please specify the password.
密码:
230 Login successful.
ftp> ls -l                                      #查看根目录内容
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Transfer done (but failed to open directory).
ftp>
```

从以上的测试过程可以看出，两个 FTP 虽然在同一台服务器中，但两者之间是完全独立的，管理员可以对每个 FTP 进行单独的配置。

9.3.4 虚拟用户的配置

在之前的章节中，我们已经介绍过，VsFTPd 的用户来源有本地用户和匿名用户两种。本地用户为操作系统中创建的用户，用户名和密码存放在系统中的/etc/passwd 文件中。这些账户既可以登录操作系统，也可以作为 FTP 用户登录到 VsFTPd 中。而匿名用户可以使用任意字符作为密码登录 VsFTPd，其权限对应操作系统中 FTP 账户的权限。

除了以上两种用户来源外，VsFTPd 还支持虚拟用户，它的用户名和密码存放在特定的数据文件中，只有在 VsFTPd 中才有效，而不能在其他系统中登录。

与匿名用户和本地用户相比，采用虚拟用户有很多优点：

- 有利于操作系统安全，FTP 账户的泄露不会对操作系统的安全造成影响。
- 如果 FTP 中需要很多账号，不需要在系统中逐个建立，减轻了操作系统的管理负担。
- 虚拟用户的账号权限设置更加方便，存储位置更加灵活。

在实际应用中，我们经常使用 PAM 作为认证程序为 VsFTPd 的虚拟用户服务。PAM(Plugable Authentication Module，可插拔认证模块)是一种完成通用认证功能的程序，其优点在于可以被其他程序调用。当使用 PAM 时，调用程序不需要重新编译，只需要通过添加一个配置文件来决定认证模块如何插入到程序中。

PAM 定义了 4 种类型的模块：

- auth 模块：提供了实际的认证过程，即提示输入并检查密码的正确性。
- account 模块：负责检查并确认是否可以进行认证。例如，账号是否到期，用户是否已经登录等。
- password 模块：主要用来修改用户的密码。
- session 模块：提供对会话的管理和认证。

下面，我们就以 PAM 为例，介绍 VsFTPd 虚拟用户的配制方法。

(1) 正常启动 VsFTPd，使用默认的 vsftpd.conf。

(2) 建立用户列表文件，创建 ftpuser.txt 文件，其中每一行分别对应一个用户名及其密码，内容如下：

```
user1
passwd1
user2
passwd2
user3
passwd3
```

(3) 安装 PAM 数据库应用程序。

首先，我们要确保 DB 及其工具包已经正常安装，如果没有正确安装，我们需要手动安装 db4-utils 和 db4-devel 应用程序。

```
[root@CentOS /]# rpm -qa | grep db4           #查看 DB 工具包
db4-4.3.29-10.el5                              #缺少 db4-utils 和 db4-devel
[root@CentOS /]# mount /dev/cdrom /mnt/cdrom   #装载安装光盘
[root@CentOS /]# cd mnt/cdrom
[root@CentOS cdrom]# cd CentOS/
[root@CentOS CentOS]#
[root@CentOS CentOS]# ls -f db4*               #查看光盘中可用的安装包
db4-4.3.29-10.el5.i386.rpm       db4-tcl-4.3.29-10.el5.i386.rpm
db4-devel-4.3.29-10.el5.i386.rpm  db4-utils-4.3.29-10.el5.i386.rpm
db4-java-4.3.29-10.el5.i386.rpm
[root@CentOS CentOS]# rpm -ivh db4-utils-4.3.29-10.el5.i386.rpm
                                                        #安装 db4-utils
Preparing...          ########################################## [100%]
  1:db4-utils         ########################################## [100%]
[root@CentOS CentOS]# rpm -ivh db4-devel-4.3.29-10.el5.i386.rpm
                                                        #安装 db4-devel
Preparing...          ########################################## [100%]
  1:db4-devel         ########################################## [100%]
[root@CentOS CentOS]# cd /
[root@CentOS /]# rpm -qa | grep db4                     #再次查看安装包
db4-devel-4.3.29-10.el5
db4-utils-4.3.29-10.el5
db4-4.3.29-10.el5
[root@CentOS /]# umount /dev/cdrom
```

(4) 生成虚拟账号数据库。

db4 的程序包正确安装后，使用以下命令在/etc/vsftpd 目录中生成账号数据库文件 ftpuser.db，并设置访问权限为 600。

```
[root@CentOS /]# db_load -T -t hash -f ./usr/ftpuser.txt
/etc/vsftpd/ftpuser.db
[root@CentOS /]# chmod 600 /etc/vsftpd/ftpuser.db
```

(5) 设置 PAM 配置文件。

所有支持 PAM 的程序都会有一个与 PAM 进行对接的配置文件，存放在/etc/pam.d 目录中，VsFTPd 与 PAM 的对接配置文件名由 vsftpd.comf 中的 pam_service_name 选项来确定，默认名为 ftp。除此之外，我们还需要在/etc/pam.目录中创建 vsftpd_login 文件，其内容如下：

```
auth required /lib/security/pam_userdb.so db=/etc/vsftpd/ftpuser
account required /lib/security/pam_userdb.so db=/etc/vsftpd/ftpuser
```

这两个语句是通过调用 pam_userdb.so 模块完成账户认证的，在认证过程中使用/etc/vsftpd/ftpuser 提供的信息进行认证。

(6) 建立 FTP 虚拟用户对应的操作系统账号，并设置该账号工作目录的权限。

为了让此账号只用作 FTP 的虚拟用户账号，我们添加了“-s /shin/login”选项，这意味此账号永远不能在操作系统中登录。

```
[root@CentOS /]# useradd -d /home/ftpvir -s /sbin/nologin ftp_vir
[root@CentOS /]# chmod 700 /home/ftpvir
```

(7) 在 vsftpd.conf 中添加有关虚拟用户的配置选项。

```
guest_enable=YES
guest_username=ftp_vir
pam_service_name=vsftpd_login
```

(8) 设置虚拟用户的权限。

VsFTPd 可以为每一个虚拟用户设置单独的配置文件，其文件名和用户名相同，这些配置文件需要统一放在一个目录下，目录位置由 uesr_config_dir 选项指定，这样，每个虚拟用户就可以在自己的配置文件中设置不同的内容，实现不同用户拥有不同权限的功能。

首先，我们必须在 vsftpd.conf 中添加 uesr_config_dir 选项，指定配置文件的存放位置。

```
uesr_config_dir=/etc/vsftpd
```

然后在/var/ftp 中为用户创建属于这个 FTP 用户的根目录。

```
[root@CentOS /]# mkdir /var/ftp/user1
[root@CentOS /]# mkdir /var/ftp/user2
```

为了使 ftp_vir 账户对这两个目录拥有完全的权限，还需要执行以下的命令：

```
[root@CentOS /]# chown ftp_vir  /var/ftp/user1
[root@CentOS /]# chown ftp_vir  /var/ftp/user2
```

下面，我们为开始创建的 3 个用户分别建立配置文件，并填写配置信息。

```
[root@CentOS /]# vi /etc/vsftpd/user1
local_root=/var/ftp/user1                  #设置 user1 工作目录
[root@CentOS /]# vi /etc/vsftpd/user2
local_root=/var/ftp/user2                  #设置 user2 工作目录
[root@CentOS /]# vi /etc/vsftpd/user3
local_root=/var/ftp/user2
anon_mkdir_write_enable=YES           #开启 user3 对 user2 目录的控制权限
anon_other_write_enable=YES
anon_upload_enable=YES
anon_world_readable_only=YES
write_enable=YES
```

以上步骤完成后，再次重启 VsFTPd 服务，所有设置就可以自动生效了，下面我们对刚才的设置进行测试，看看是否和我们预想的一样。

```
C:\Users\Administrator>ftp 192.168.111.133
连接到 192.168.111.133。
220 (vsFTPd 2.0.5)
用户(192.168.111.133:(none)): user3      #使用 user3 登录
331 Please specify the password.
密码:
230 Login successful.
ftp> mkdir user3                         #创建目录
```

```
257 "/user3" created                          #目录创建成功
ftp> ls -l                                     #查看创建的目录
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwx------    2 501      501          4096 Feb 23 08:07 user3
226 Directory send OK.
ftp: 收到 63 字节，用时 0.00 秒 63000.00 千字节/秒。
ftp> close
221 Goodbye.
ftp> open 192.168.111.133
连接到 192.168.111.133。
220 (vsFTPd 2.0.5)
用户(192.168.111.133:(none)): user2        #使用 user2 登录
331 Please specify the password.
密码:
230 Login successful.
ftp> ls -l
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwx------    2 501      501          4096 Feb 23 08:07 user3
226 Directory send OK.
ftp: 收到 63 字节，用时 0.00 秒 63.00 千字节/秒。
ftp> rmdir user3                               #试图删除 user3 创建 user3 的目录
550 Permission denied.                         #删除失败，没有权限
ftp> close
221 Goodbye.
ftp> open 192.168.111.133
连接到 192.168.111.133。
220 (vsFTPd 2.0.5)
用户(192.168.111.133:(none)): user1          #使用 user1 登录
331 Please specify the password.
密码:
230 Login successful.
ftp> ls -l                        #查看根目录内容，明显与 user2 处于不同的目录
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
ftp> mkdir user1                               #试图创建目录
550 Permission denied.                         #创建失败，没有权限
ftp> bye
221 Goodbye.
```

经过以上的测试，可以看出我们创建的虚拟用户都能够正常的登录到 VsFTPd 中，并且所有用户的权限与设计相符。

9.3.5　FTP 日志的配置

在 VsFTPd 中，日志可以记录服务器运行期间的各种工作状态，包括用户登录、文件下载、创建目录等操作。有了日志的帮助，管理员可以及时了解服务器运行中出现的各种问题，也可以在服务调试的过程中了解相关信息。

1. 开启日志

要开启 VsFTPd 中的日志功能，只需要在 vsftpd.conf 中添加下面的选项：

```
xferlog_enable=YES
```

即可开启日志功能，默认情况下，日志保存在/var/log 目录中的 vsftpd.log 文件中，如果管理员想更改日志文件的位置或者文件名，可以在 vsftpd.conf 中添加以下选项：

```
vsftpd_log_file=路径及文件名
```

vsftpd.log 中的日志格式是 VsFTPd 特有的，具有良好的可读性。另外，管理员还可以使用以下选项将日志的格式输出为 xferlog 格式：

```
xferlog_std_format=YES
```

xferlog 格式是与 wu-FTP 服务器兼容的一种通用 FTP 日志格式，可读性比较差，但是很多日志分析软件可以直接对这种格式进行分析。xferlog 日志默认记录在/var/log 目录的 xferlog 文件中，它的位置和名称也可以通过以下选项进行修改：

```
xferlog_file=路径及文件名
```

默认情况下，VsFTPd 只记录其特有格式的日志，如果希望同时还记录 xferlog 格式的日志，需要在 vsftpd.conf 中添加一遍以下选项，让系统同时记录两种格式的日志，分别储存在对应的路径文件中。

```
dual_log_enable=YES
```

2. VsFTPd 日志格式

下面是一些记录在 vsftpd.log 文件中的日志片段。

```
Tue Feb 21 02:21:34 2012 1 192.168.111.133 0 /ftpfile1 a _ o a 123 ftp 0 * i
Tue Feb 21 02:23:31 2012 1 192.168.111.1 0 /ftpfile1 a _ o a 123 ftp 0 * i
Tue Feb 21 02:25:26 2012 1 192.168.111.1 0 /upload/ntuser.dat.LOG1 a _ i
a 123 ftp 0 * i
Tue Feb 21 02:32:30 2012 1 192.168.111.1 0 /upload/upfile.txt a _ i a
123 ftp 0 * i
Tue Feb 21 02:34:26 2012 1 192.168.111.1 18 /upload/upfile.txt a _ o a
123 ftp 0 * c
```

以上日志中，每一列的具体含义如下：

- 记录日志的时间。
- 文件传输所使用的时间。
- 客户端的名称和 IP 地址。
- 传输的字节数。
- 上传或者下载的文件名及其路径。
- 传输的方式：a 代表 ASCII 方式；b 代表二进制方式。
- 行为标志：“_”表示没有行为；其余的保留未用。
- 传输方式：相对服务器而言，o 表示输出(out)；i 表示输入(in)。
- 访问方式：a 表示匿名用户(anonymous)；g 表示访客(guest)；r 表示真实系统用户(real)。
- 用户名：访问 FTP 的用户名。
- 服务器名：都为 FTP。

- 认证方式：0 表示未使用。
- 认证的用户 ID：*表示未使用。
- 完成标志：c 表示完成传输；i 表示未完成传输。

9.3.6　磁盘限额的配置

对于 FTP 服务器而言，其存储空间总是有限的。但是，每个用户都会倾向于使用更多的磁盘空间，为了避免磁盘空间被很快耗尽而不能正常提供服务的情况发生，大多数 FTP 的管理员都会选择对用户可使用的磁盘空间进行限制。下面我们就介绍一下使用 quota 软件对用户可使用的磁盘空间进行限制的方法。

1. 设置分区支持磁盘限额

磁盘限额工具(Quota)是系统管理员用来监控和限制用户或者组对磁盘的使用的工具。磁盘限额只能针对某个分区而言，也就是说，一个磁盘分区要么全部用于限制用户使用，要么全部不参与磁盘限额。另外，如果要使用磁盘限额，还需要保证在 Linux 内核编译的过程中必须设定支持 quota，这一点在 CentOS5.5 中已经做到。

提示： 磁盘限额是针对 Linux 系统中的一般用户而言，root 权限用户不会受到磁盘限额的限制。

quota 的限额方法有两种：

- 限制用户或组可以拥有的 inode 数，即文件数。
- 限制用户或组可使用的磁盘块数，即磁盘空间。

要使用 quota 进行磁盘限额，首先要保证软件包正确的安装，使用以下的命令来查看 quota 软件包的安装情况。

```
[root@CentOS /]# rpm -qa | grep quota
quota-3.13-1.2.5.el5
```

从显示内容我们可以看出，quota 软件已经正常安装。为了使用磁盘限额的功能，首先需要设置挂载文件系统时对磁盘限额的支持，我们可以通过修改/etc/fatab 文件来实现，在其中需要进行磁盘限额的分区中添加 usrquota(用户限额)或者 grpquota(组限额)选项，如下面的内容：

```
[root@CentOS etc]# more fstab
/dev/VolGroup00/LogRoot /                ext3    defaults        1 1
/dev/VolGroup00/Home    /home            ext3    defaults,usrquota,grpquo
ta       1 2            #在/home 分区中开启对用户限额和组限额的支持
LABEL=/boot             /boot            ext3    defaults        1 2
tmpfs                   /dev/shm         tmpfs   defaults        0 0
devpts                  /dev/pts         devpts  gid=5,mode=620  0 0
sysfs                   /sys             sysfs   defaults        0 0
proc                    /proc            proc    defaults        0 0
/dev/VolGroup00/LogVol01 swap            swap    defaults        0 0
```

设置分区对磁盘限额的支持后，我们需要重启服务器或者使用以下命令重新挂载分区。

```
[root@CentOS /]# mount -o remount /home
```

之后，我们还需要在此分区中使用 quotacheck 命令来自动创建两个文件，分别用来记录用户和组在该分区磁盘空间的使用情况，如下面的命令所示：

```
[root@CentOS home]# ls
ftpvir  lost+found  mvdir
[root@CentOS home]# quotacheck -agu
[root@CentOS home]# ls
aquota.group  aquota.user  ftpvir  lost+found  mvdir
```

此命令中，“-a”选项表示扫描所有的挂载分区文件/etc/fstab，如果发现该挂载分区中有 usrquota 选项，那么会在其分区中创建 aquota.user 文件，用来记录用户在此分区的磁盘使用情况。“-g”与“-a”类似，用来检查分区文件中是否有 grpquota 选项，如果有，则在此分区根目录中创建 aquota.group 文件，用来记录组用户在此分区的磁盘使用情况。而“-u”是默认选项，用来在创建文件前先检查是否已经有同名文件，这样做主要是为了防止在之前已经创建磁盘配额的分区中将原有的文件覆盖。

2. 设置用户磁盘限额

进行了以上的配置后，系统已经能够针对/home 分区对用户进行磁盘限额的设置，方法是通过使用 quota 软件包中的 edquota 命令来实现。此命令的使用格式如下：

```
edquota [选项] [-p 用户名] [-f 文件系统] <用户名 | 用户组>
```

edquota 命令可用的选项如表 9-2 所示。

表 9-2 quota 选项及其含义

选 项	含 义
-u	对用户进行磁盘限额的设置，默认设置
-f [文件系统]	用于对指定的磁盘分区设置限额
-p 用户名或者用户组名	以某一个用户为模板进行用户的磁盘限额设置
-g	对用户组进行磁盘限额的设置
-t	对所有用户或者用户组进行宽限时间限制
-T	对选定的用户或者用户组进行宽限时间限制

提示： 磁盘限额是针对 Linux 系统中的一般用户而言，root 权限用户不会受到磁盘限额的限制。在 quota 中，磁盘空间限额方式有两种：

① 软限额：用户到达限额后，系统将给出警告，但用户在宽限时间内还可以继续使用超额的空间或者 inodes。

② 硬限额：用户将永远不能超过这个限额。

宽限时间是指当管理员使用软限额时，用户的磁盘限额超过软件额后，要求在多长时间内必须将使用额度下降到软件额之下，否则将不能继续使用空间或者 inode 数。

例如，我们使用 edquota ftp 命令来修改 FTP 用户的磁盘限额，此时实际上 edquota 将调用 vi 编辑器对用户的配置进行编辑，显示内容如下：

```
Disk quotas for user ftp (uid 14):
  Filesystem              blocks    soft     hard     inodes     soft     hard
  /dev/mapper/VolGroup00-Home  0       0        0         0        0        0
```

以上内容从左至右各个字段的含义如下：

- Filesystem：限额的分区。
- blocks：表示该用户已经使用的磁盘块数。
- soft：磁盘块数的软限制。
- hard：磁盘块数的硬限额。
- inodes：用户已经使用的 inode 数。
- soft：inode 数的软限额。
- hard：inode 数的硬限额。

我们可以修改此文件中的各个数字来调整其软限额和硬限额，例如下面的内容表示 FTP 用户在磁盘分区/dev/mapper/VolGroup00-Home 中最多只能使用 100 个磁盘块，如果达到 90 个磁盘块将会给出警告，而且最多只能创建 5 个文件，达到 4 个的时候将给出警告。

```
Disk quotas for user ftp (uid 14):
  Filesystem              blocks    soft     hard     inodes     soft     hard
  /dev/mapper/VolGroup00-Home  0       90       100       0        4        5
```

默认情况下，每个磁盘块代表 1KB 的容量，但并不能说 100 个磁盘块就能够存储 100KB 的文件内容，因为很多情况下并不是所有的磁盘块都能够被全部利用。

如果管理员要对宽限时间进行设置，可以使用以下的命令，其显示内容如下：

```
[root@CentOS ~]# edquota -t
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
  Filesystem                    Block grace period    Inode grace period
  /dev/mapper/VolGroup00-Home           7days                   7days
```

从显示的内容我们可以看出，此时对所有用户的宽限时间为 7 天，除了使用 days 外，还可以使用 hours、minutes、seconds 等单位。

3. 开启磁盘限额

设置好所有用户和用户组的磁盘限额后，就可以开启磁盘限额功能，使用以下的命令格式来开启指定文件系统的磁盘限额功能。

```
quotaon [文件系统]
```

或者，我们也可以使用以下命令来开启所有已经设置了磁盘限额的分区。

```
quotaon -aguv
```

4. 关闭磁盘限额

如果要关闭磁盘限额功能，可以使用以下命令：

```
quotaoff
```

5. 测试实例

下面，我们以一个实例的形式对刚才所讲述的内容进行测试及验证。测试实例如下：

```
[root@CentOS /]# cd home
[root@CentOS home]# ls
aquota.group  aquota.user  ftpvir  lost+found  mvdir
[root@CentOS home]# chown ftp mvdir              #改变 mvdir 的权限
[root@CentOS home]# quota ftp                    #查看 FTP 账户的磁盘限额情况
Disk quotas for user ftp (uid 14):
  Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
/dev/mapper/VolGroup00-Home
                          4      90     100               1       4       5
```

9.4 FTP 客户端的配置

使用客户端访问 VsFTPd 服务器比较简单，可以使用各种 FTP 客户端软件来实现，包括图形化工具和命令行工具。

1. 使用图形化工具访问 VsFTPd 服务器

我们可以使用 Web 浏览器来访问 VsFTPd 服务器，一般浏览器都具有这样的功能。例如 Internet Explorer、Mozilla 或者 Firefox 浏览器等。

如果用户使用匿名访问，可以直接在地址栏中输入 ftp://FTP 服务器的名称或者 IP 地址，如图 9-2 所示；如果使用本地用户或者虚拟用户，需要在地址栏中输入 ftp://用户名:密码@FTP 服务器的名称或者 IP 地址。



图 9-2 使用 IE 访问 VsFTPd 服务器

除了使用 Web 浏览器外，还可以使用其他专业的 FTP 客户端软件来访问 VsFTPd，例如 cuteFTP、FlashFXP 等。

2. 使用命令行访问 VsFTPd 服务器

不管在 Windows 系统还是 Linux 系统，其命令行工具都带有 FTP 命令工具，用户可以直接使用 FTP 命令来访问 VsFTPd 服务器。如图 9-3 是使用 Linux 中的命令行来访问 VsFTPd 的过程。

在 FTP 中常用的命令及其含义如表 9-3 所示。

另外，在使用命令行工具访问 VsFTPd 服务器时，系统将以代码的形式表示出服务器的不同状态。通过这些代码，用户可以判断服务器工作是否正常，以对 FTP 服务器进行调试或者故障诊断。表 9-4 是 FTP 服务器常见的代码及其含义。

```
[root@CentOS ~]# ftp 192.168.44.133
Connected to 192.168.44.133.
220 (vsFTPd 2.0.5)
530 Please login with USER and PASS.
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
Name (192.168.44.133:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -l
227 Entering Passive Mode (192,168,44,133,117,133)
150 Here comes the directory listing.
-rw-------    1 0        0               0 Feb 21 02:14 ftpfile1
-rw-------    1 0        0               0 Feb 21 02:14 ftpfile2
drwxr-xr-x    2 0        0            4096 Feb 21 02:33 upload
226 Directory send OK.
ftp> cd upload
250 Directory successfully changed.
ftp> _
```

图 9-3　使用命令行访问 VsFTPd 服务器

表 9-3　FTP 常用命令及其含义

命　令	含　义
! command	执行本地的 shell 命令
ls	列出文件盒目录清单
dir	与 ls 命令相同
pwd	显示 FTP 服务器端的工作目录
cd	改变当前目录
lcd	查看或者改变本地的工作目录
get	从 FTP 下载单个文件
put	向 FTP 上传单个文件
mget	支持通配符，从 FTP 下载多个文件
mout	支持通配符，向 FTP 上传多个文件
mkdir	创建目录
rm	删除文件或者目录
passive	打开或者关闭 PASV 模式
close	退出当前 FTP 会话
bye	关闭 ftp 命令

表 9-4　FTP 服务器常见代码及其含义

代　码	含　义
110	重新启动标志回应
120	服务在多长时间内可用
125	数据连接已经打开，开始传输数据
150	文件状态正确，正在打开数据连接
200	命令执行正常结束
202	命令未被执行
211	系统状态或者系统帮助信息回应

续表

代　码	含　义
212	目录状态信息
213	文件状态信息
214	帮助信息
215	NAME 系统类型
220	新连接的用户的服务已经就绪
221	控制连接关闭
225	数据连接已打开，没有进行中的数据传递
226	正在关闭数据连接，请求文件动作成功结束
227	进入被动模式
230	用户已经登录
250	被请求文件操作成功完成
257	路径已经建立
331	用户名存在，需要输入密码
332	需要登录的账户信息
350	对被请求文件的操作需要进一步更多的信息
421	服务不可用，控制连接关闭
425	打开数据连接失败
426	连接关闭，传输中止
450	对被请求文件的操作没有执行
452	请求的操作没有被执行，系统存储空间不足或者文件不可用
500	语法错误，不可识别的命令
501	参数错误导致的语法错误
502	命令未被执行
503	命令的次序错误
504	由于参数错误，命令未被执行
530	没有登录
532	存储文件需要账户信息
550	请求操作未被执行，文件不可用
551	请求操作终止，页面类型位置
552	对请求文件的操作中止，超出存储分配
553	请求的操作没有执行，文件名不允许

9.5　VsFTPd 综合案例

经过上面的介绍，我们已经能够比较详细的了解 vsftpd.conf 的各项参数了。本节中，

我们将通过一个比较详细的实例来实现 VsFTPd 的安全匿名访问，具体操作步骤如下：

(1) 使用 vi 编辑器打开/etc/vsftpd/vsftpd.conf。

(2) 所有的配置内容如下面的代码，关键内容都已经给出了解释。

```
listen=YES
listen_address=192.168.44.133

anonymous_enable=YES       #开启匿名访问
local_enable=NO            #关闭本地账户访问

write_enable=NO              #取消匿名用户的写权限
anon_upload_enable=YES
anon_mkdir_write_enable=NO
anon_other_write_enable=NO
anon_world_readable_only=YES    #只允许匿名用户浏览文件

xferlog_enable=YES          #开启日志记录
xferlog_file=/var/log/xferlog

connect_from_port_20=YES     #对用户设置超时时间，进行连接控制
pasv_min_port=50000
pasv_min_port=60000

max_clients=100             #控制最大并发数，限定每个 IP 并发数
max_per_ip=10

anon_max_rate=80000

ftpd_banner=this is offical.     #设置欢迎信息

hide_ids=YES                #隐藏文件的所有者和组信息，匿名用户看到的所有文件的所有者
和组都变为 ftp
```

9.6　本 章 小 结

本章从 FTP 的工作原理讲起，详细介绍了 VsFTPd 服务器的配置过程。学习本章内容的重点是 vsftpd.conf 文件的参数设置。

9.7　课 后 习 题

1. 填空题

(1) FTP 是____________________协议的缩写。

(2) FTP 的连接模式分为__________和__________两种。

2. 选择题

(1) FTP Server 通常使用(　　)服务通道。

A. 13　　B. 25　　C. 8　　D. 21

(2) port 20 通常是(　　)服务通道。

A. web data　　B. ftp data　　C. mail data　　D. dns data

(3) 在使用匿名登录 FTP 时，用户名为(　　)。

A. users　　B. anonymous　　C. root　　D. guest

3. 简答题

(1) 如何启动 VsFTPd 服务？它的主要配置文件是什么？

(2) FTP 启用的两个端口分别是什么？这两个端口在哪里设置？默认的端口是多少？

4. 操作题

架设一个用于朋友之间互传文件的 FTP 服务器。要求只能使用匿名登录，匿名用户有读写权限，但不能离开 FTP 根目录。

第 10 章

Mail 服务的配置及应用

电子邮件(Electronic Mail，E-mail)是通过网络实现相互传送和接收信息的现代化通信方式，是互联网中最基本、最常用的服务之一。用户可以通过它实现与远程用户经济、方便、快捷的信息交流。本章在介绍电子邮件原理的同时以 postfix 服务为例介绍电子邮件系统的安装、配置和使用。

10.1 电子邮件服务概述

电子邮件服务是 Internet 最基本的服务，也是最重要的服务之一。与传统的邮政信件服务类似，电子邮件可以用来在 Internet 或 Intranet 上进行信息的传递和交流，具有快速、经济的特点。与其他各种网络服务相比，电子邮件服务相对比较复杂，要涉及多种网络协议，而且一个实际的邮件系统往往由很多相互独立的软件包组成，还需要解决它们之间集成时接口的兼容问题。在本节中，我们先对邮件系统的组成和工作原理进行介绍。

10.1.1 邮件系统的组成及工作原理

与其他网络服务相同，电子邮件服务是基于客户/服务器(C/S)模式的。但邮件从发件人客户端到达收件人客户端的过程中，还需要邮件服务器之间的相互传输，因此，邮件系统要比其他的网络服务(例如 FTP、Web 等)复杂得多。基本的传输流程如图 10-1 所示。

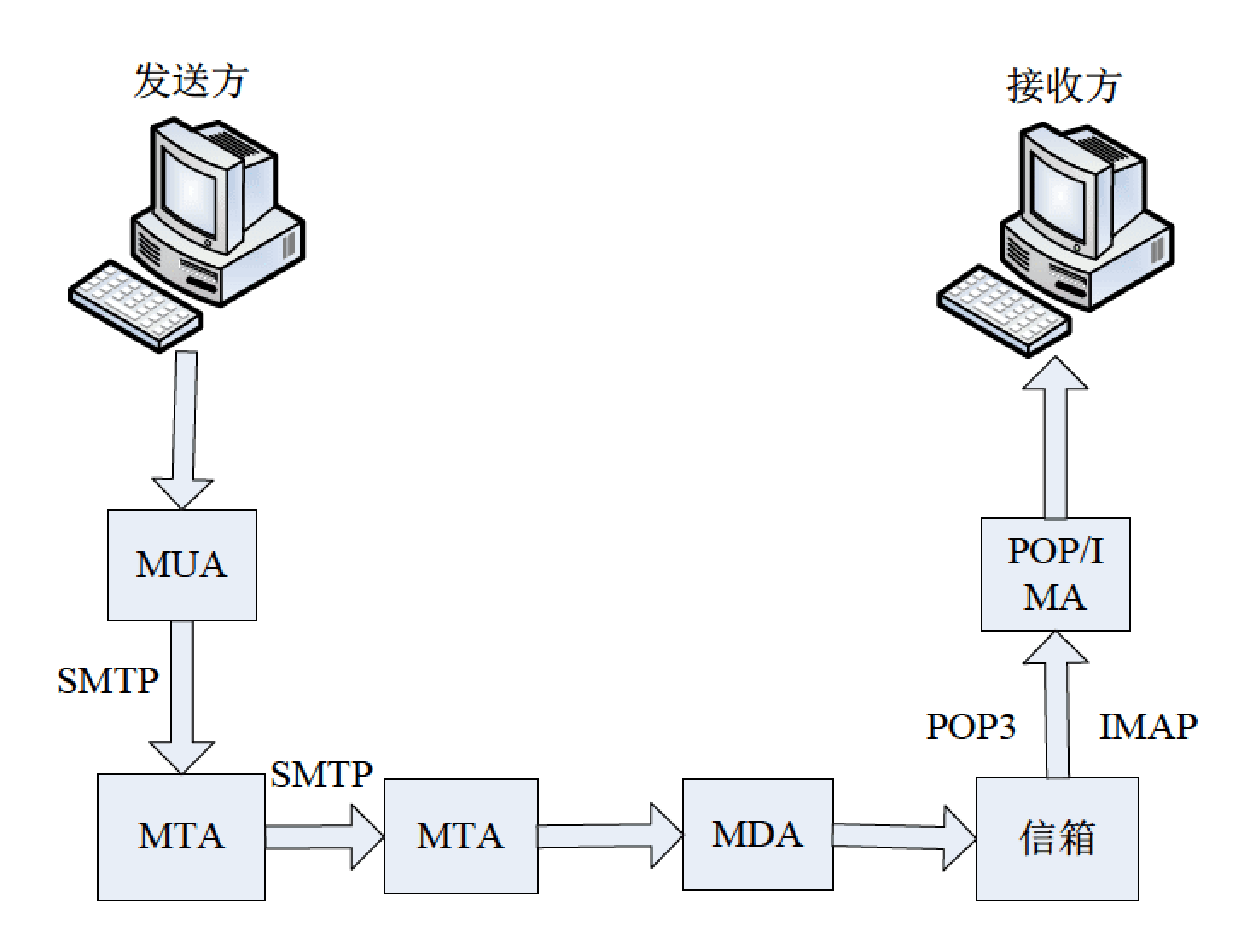


图 10-1 邮件传输流程示意图

其中主要用到的协议和软件如下。

1. 邮件用户代理(MUA)

邮件用户代理(Mail User Agent)是在邮件使用终端上运行的程序，主要负责编辑和发送邮件，以及从服务器上下载、管理、阅读和处理邮件。目前常用的邮件用户代理 Windows 平台主要有 Outlook、Foxmail、Dreammail 等，Linux 平台主要有 Evolution、Thunderbird、KMail 等。

2. 邮件传输代理(MTA)

邮件传输代理(Mail Transfer Agent，MTA)主要用于存储和发送邮件，也可以说是邮件服务器软件的总称，如 Sendmail、Postfix、Qmail、Exim 等。一台服务器可以安装多个

MTA，但同一时刻只能有一个 MTA 工作。

3. 邮件分发代理(MDA)

邮件分发代理(Mail Delivery Agent，MDA)主要负责将 MTA 接收的邮件传递到收件人的邮箱(Mailbox)中。所有的 SMTP 服务器也都可以称为 MDA，但并非所有的 MDA 都是 SMTP 服务器。

4．电子邮件使用的协议

要实现电子邮件服务还必须借助于专用的协议才行。目前，应用于电子邮件服务的协议主要有 SMTP、POP3 和 IMAP4 协议。

1) SMTP 协议

SMTP 即简单邮件传输协议，它是一组用于由源地址到目的地址传送邮件的规则，由它来控制信件的中转方式。SMTP 协议属于 TCP/IP 协议簇，它帮助每台计算机在发送或中转信件时找到下一个目的地。

SMTP 协议的一个重要特点是它能够以接力的方式传输邮件，即邮件可以通过不同的网络主机一站接着一站地传输。SMTP 协议属于请求/应答式范式，请求和应答都属于 ASCII 文本，并以 CR 和 LF 符结束。应答包含一个 3 位数字的代码，以及供人阅读的文本解释。常见的 SMTP 命令如表 10-1 所示。

表 10-1　SMTP 常用命令及其含义

命　令	含　义
HELLO 客户机域名	鉴别对方是否支持 SMTP 协议，应该作为发送方的第一个命令
EHLO	鉴别接收方是否支持 ESMTP 协议，接收方将返回所有支持的扩展命令
AUTH	开始进行认证
MAIL FROM 发件人地址	告诉接收方即将发送一封新邮件，并对所有的状态和缓冲区进行初始化
RCPT TO 收件人地址	标识各种邮件接收人地址，该命令可以发送多个，表示有多个收件人
DATA	告诉接受方此后为邮件正文，直到以“.”为唯一内容的一行为止
REST	退出 /复位当前的邮件传输
NOOP	空操作，用于保持 TCP 连接
QUIT	要求停止传输并关闭 TCP 连接
VRFY 字符串	验证给定的邮箱是否存在，出于安全考虑，大多数 SMTP 服务器都会禁止该命令
EXPN 字符串	查询是否有邮箱属于给定的邮件列表，出于安全考虑，大多数 SMTP 服务器都会禁止该命令
DEBUG	告知接收方开始调试，接收方将处于调试状态
HELP	返回帮助信息

SMTP 接收方收到命令后，将根据具体情况给发送方返回应答，应答是以应答码的形式发送的，所有应答码及其解释如表 10-2 所示。

表 10-2 SMTP 应答码及其含义

应 答 码	含 义
200	表示成功执行了命令，并不是标准的应答
211	系统状态或系统帮助回复
214	帮助信息
220	“域名称”服务已准备就绪
221	“域名称”服务正在关闭传输通道
250	所请求的命令已经成功执行
354	开始输入邮件
421	“域名称”服务无效，正在关闭传输通道
450	因邮箱无效，所请求的 MAIL 命令没有执行
451	因本地处理错误，放弃执行所有请求的命令
452	因系统存储空间不足，所请求的命令没有执行
500	语法错误，命令没有执行
501	命令的参数存在语法错误
502	无效的命令
503	不正确的命令次序
504	无效的命令参数
521	“域名称”不接受邮件
530	拒绝访问
550	因邮箱无效，所请求的 MAIL 命令无效
551	非本地用户
552	因超过存储分配，MAIL 命令无效
553	因邮箱名不允许，请求的命令没有执行
554	传输事务失败

下面，我们以一个典型的客户端与邮件服务器建立连接的过程为例，介绍双方的交互过程。

(1) 每次发送邮件时，用户代理需要与邮件所在的 SMTP 服务器建立 TCP 连接，然后服务器会发送给客户端 220 应答报文。

(2) 接着客户端发送 EHLO 命令，服务器回应 250 应答报文，表示服务器支持扩展的 SMTP 命令，并处于就绪状态。

(3) 客户端发送 AUTH LOGIN 命令，告诉服务器开始进行验证，服务器回应 334 报文，要求开始输入用户名。

(4) 客户端发送经过编码的用户名给服务器，服务器在此发送要求输入密码，客户端会再次将经过编码的密码发送给服务器，服务器回应 235 报文，表明验证成功。

(5) 此时客户端可以开始发送邮件，首先发送 MAIL 命令将发件人地址告诉服务器，再通过 RCPT 命令发送收件人地址，之后开始发送 DATA 命令及其邮件内容，当然，这期间每一个命令服务器都会回应 250 报文表示命令执行成功。

(6) 所有邮件内容发送完毕后，客户端发送 QUIT 命令告诉服务器退出，服务器回应 221 报文关闭传输通道。

(7) 最后，拆除客户端和服务器之间的 TCP 连接。

以上的过程是邮件客户端发送邮件到 SMTP 的过程，之后，SMTP 服务器会以类似的过程将邮件再转发到收件人所在的 SMTP 服务器，但是 SMTP 服务器之间传递邮件时，不需要使用 AUTH LOGIN 命令进行认证。

2) POP3 协议

邮件客户端通过 SMTP 协议将邮件传递给服务器，但邮件客户端读取信件时，使用的是另一种协议——POP3(Post Office Protocol 3)，即邮局协议的第 3 个版本，它规定怎样将个人计算机连接到 Internet 的邮件服务器和下载电子邮件的协议。它是 Internet 电子邮件的第一个离线协议标准，POP3 允许从服务器上把邮件存储到本地主机即自己的计算机上，同时删除保存在邮件服务器上的邮件。遵循 POP3 协议来接收电子邮件的服务器是 POP3 服务器。

POP3 协议也是建立在 TCP 协议基础上的应用层协议，默认使用 110 端口。POP3 协议与 SMTP 类似，也属于请求应答范式。客户端首先向 POP3 服务器的 110 端口发起 TCP 连接请求，服务器接收后，双方建立 TCP 连接。之后，客户端向服务器发送 POP3 命令，服务器收到命令后，根据具体情况决定是否执行，然后回复相应的应答。常用的 POP3 命令及其含义如表 10-3 所示。

表 10-3　POP3 常用命令及其含义

命　令	含　义
USER 用户名	提交用户名
PASS 密码	提交密码
STAT	请求服务器返回信箱统计信息，如邮件数，邮件大小等
LIST n	列出第 *n* 封邮件的信息
RETR n	返回第 *n* 封邮件的全部内容
DELE n	删除第 *n* 封邮件，只有 QUIT 命令执行后才会真正删除
RSET	撤销所有 DELE 命令
UIDL n	返回第 *n* 封邮件的标识
TOP n,m	返回第 *n* 封邮件的前 *m* 行内容
NOOP	空操作，用于保持 TCP 连接保持
QUIT	结束会话，退出

POP3 协议的应答非常简单，代码只有两种：“+OK”表示命令确认执行成功，“-ERR”表示错误。

下面，我们以一个典型的 POP3 协议下客户端和服务器的交互来讲述通过 POP3 协议来读取邮件的过程。

(1) 首先，客户端发送 TCP 连接请求到服务器，双方建立 TCP 连接。连接成功后，服务器返回“+OK”回应，代表已成功登录邮箱。

(2) 之后，客户端发送 USER 和 PASS 命令将用户名和密码发送给服务器，如果正确，服务器都会回应“+OK”。

(3) 客户端发送 STAT 命令查看邮件状态，服务器返回“+OK”回应并附带邮件数和邮件总字节数。

(4) 客户端此时可以对邮件进行操作，包括查看邮件(RETR)、删除邮件(DELE)等。

(5) 如果要断开连接，客户端发送 QUIT 命令要求退出，服务器回应“+OK”，然后双方拆除 TCP 连接，完成整个过程。

在以上的服务器应答中，如果客户端发送了错误的命令，服务器会回应“-ERR”应答，并且不会执行这个命令。

3) IMAP4 协议

IMAP4(Internet Message Access Protocol 4)即 Internet 信息访问协议的第 4 个版本，是用于从本地服务器上访问电子邮件的协议，它是一个客户/服务器模型协议，用户的电子邮件由服务器负责接收保存，用户可以通过浏览信件头来决定是否要下载此信。用户也可以在服务器上创建、更改文件夹或邮箱，删除信件或检索信件的特定部分。

虽然 POP 和 IMAP 都是处理接收邮件的协议，但两者在机制上却有所不同。在用户访问电子邮件时，POP3 将信件保存在服务器上，当用户阅读信件时，所有内容都会被立即下载到用户的机器上，而 IMAP4 需要持续访问服务器，因此，可以把 IMAP4 看成是一个远程文件服务器，而把 POP3 看成是一个存储转发服务器。

除了机制的不同，IMAP 还有以下几个特点：

- 在线和离线的两种操作模式。
- 用户信箱的多重连接。
- 在线浏览。
- 在服务器保存邮件的状态信息。
- 支持多信箱。
- 服务端搜索。
- 良好的扩展机制。
- 支持加密传输。

常用的 IMAP 命令如表 10-4 所示。

表 10-4 IMAP 常用命令及其含义

命　令	含　义
CREATE	创建一个新邮箱，邮箱名通常是带有路径的目录名
DELETE	删除指定名称的邮箱，邮箱名通常是带有路径的目录名，邮箱删除后，其中的邮件也一起被删除
RENAME	修改邮箱的名称，邮箱名通常是带有路径的目录名
LIST	列出邮箱的内容
APPEND	客户端上传一个邮件到指定的邮箱
SELECT	设定默认邮箱，即以后的操作都是针对此邮箱
FETCH	读取邮件的文本信息用来显示

续表

命　令	含　义
STORE	修改邮件的属性，包括设置已读标记、删除标记等
CLOSE	关闭邮箱，此时该邮箱中所有标为 DELETED 的邮件将被彻底删除
EXPUNGE	不关闭邮箱而删除所有标记为 DELETED 的邮件
EXAMINE	以制动方式打开邮箱
SUBSCRIBE	在客户端活动邮箱列表中添加一个新的邮箱
UNSUBSCRIBE	在客户端活动邮箱列表中删除一个新的邮箱
LSUB	与 LIST 命令功能类似，但只列出活动邮箱
STATUS	查询邮箱的当前状态
CHECK	在邮箱上设置一个检查点
SEARCH	根据指定的条件在处于活动状态的邮箱中搜索邮件，然后加以显示
COPY	复制邮件到其他位置
UID	邮件的唯一编号，可与其他对邮件进行操作的命令合用
CAPABILITY	请求服务器返回支持的命令列表
NOOP	空操作，用来维持 TCP 连接
LOGOUT	注销用户并关闭所有已经打开的邮箱

从 IMAP 的特征和命令含义可以看出，IMAP 协议的工作方式更适用于处理大量邮件的用户。

提示： 在实际应用中，MTA、MUA、MDA 以及 POP/IMAP 等服务器组件均可以由不同的软件来承担。另外，除了以上内容，为了方便邮件服务的管理，一个邮件服务器往往还包括账号管理、信箱管理、安全传输、提供 Web 访问界面等一系列的功能，这些功能都需要相应的软件来支持，因此，建立一个实际可用的服务器需要集成很多不同的软件。

10.1.2　主流电子邮件服务器软件

在 Linux 平台中，有许多邮件服务器可供选择，但目前使用较多的是 Sendmail 服务器、Postfix 服务器和 Qmail 服务器。

1) Sendmail 服务器

从使用的广泛程度和代码的复杂程度来讲，Sendmail 是一个很优秀的邮件服务软件。几乎所有 Linux 的缺省配置中都内置了这个软件，只需要设置好操作系统，它就能立即运转起来。但它的安全性较差，Sendmail 在大多数系统中都是以 root 身份运行的，一旦邮件服务发生安全问题，就会对整个系统造成严重影响。同时在 Sendmail 开放之初，Internet 用户数量及邮件数量都较少，使 Sendmail 的系统结构并不适合较大的负载，对于高负载的邮件系统，需要对 Sendmail 进行复杂的调整。

2) Postfix 服务器

Postfix 是一个由 IBM 资助、由 Wietse Venema 负责开发的自由软件工程产物，它的

目的就是为用户提供除 Sendmail 之外的邮件服务器选择。Postfix 在快速、易于管理和提供尽可能的安全性方面都考虑得比较周全。Postfix 是基于半驻留、互操作进程的体系结构，每个进程都要完成特定的任务，没有任何特定的进程衍生关系，能使整个系统进程得到很好的保护。同时 Postfix 也可以和 Sendmail 邮件服务器保持兼容以满足用户的使用习惯。

3) Qmail 服务器

Qmail 是由 Dan Bernstein 开发的可以自由下载的邮件服务器软件，其第一个 beta 版本 0.70.7 发布于 1996 年 1 月 24 日，当前版本是 1.03。Qmail 是按照将系统划分为不同模块的原则进行设计的，在系统中有负责接收外部邮件的模块，有管理缓冲目录中待发送邮件队列的模块，也有将邮件发送到远程服务器或本地用户的模块。同时只有必要的程序才是 setuid 程序(即以 root 用户权限执行)，这样就减少了安全隐患，并且由于这些程序都比较简单，因此就可以达到较高的安全性。

10.2 Postfix 服务及其安装

Postfix 不但在快速、易用和安全等方面都进行了较好的考虑，同时与老牌的 Sendmail 邮件服务器保持了很好的兼容性，因此是架设 Linux 平台下邮件服务器的首选。在本节中，我们将介绍 Postfix 邮件服务器的系统结构和安装运行等内容。

10.2.1 Postfix 邮件系统结构

Postfix 由十几个具有不同功能的半驻留进程组成，每个进程都提供特定的功能。但为了安全，这些进程之间又没有特定的父子进程联系。另外，Postfix 还有 4 种不同的邮件队列，由队列管理进程统一进行管理。

1. 邮件接收流程

Postfix 的邮件接收流程如图 10-2 所示。

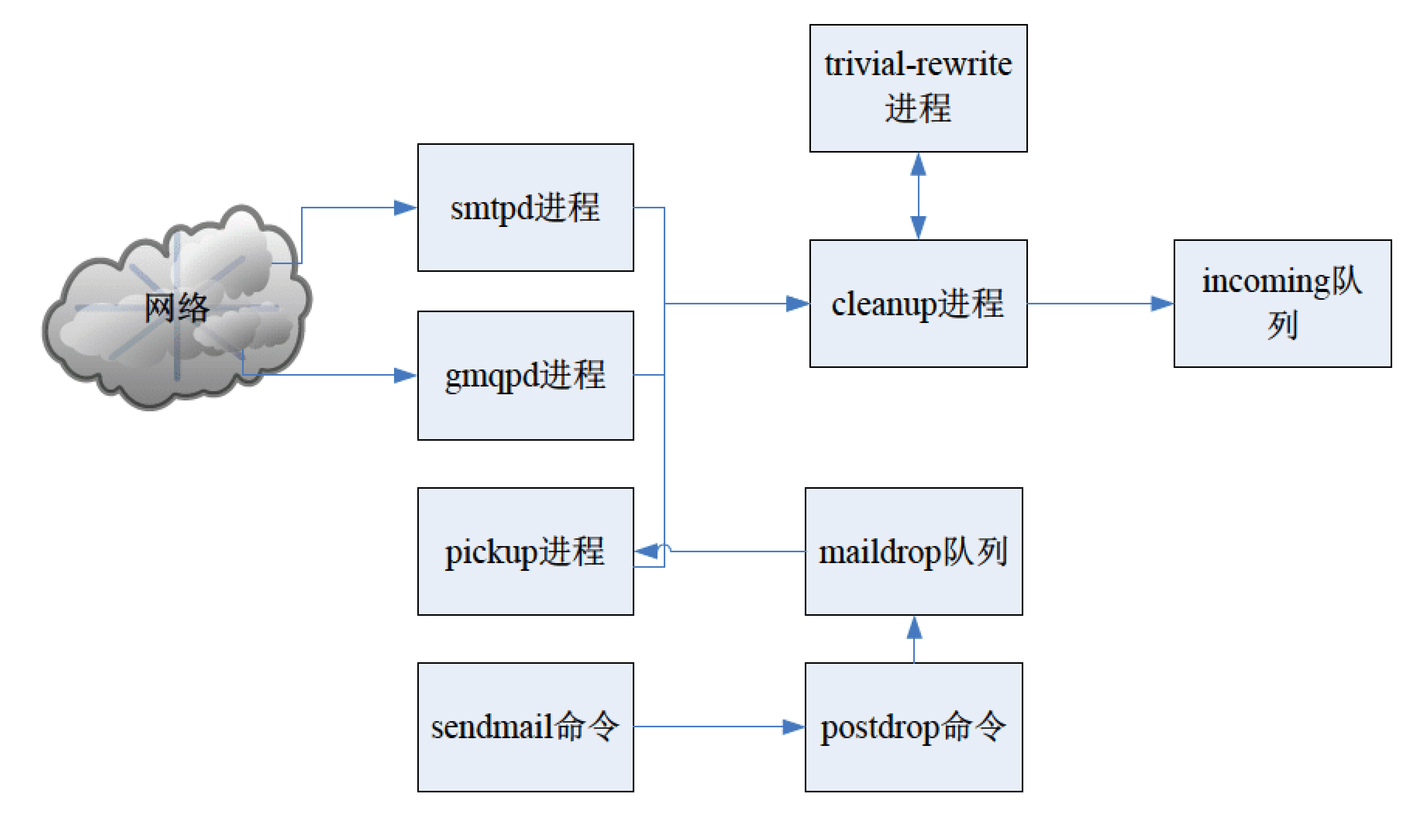


图 10-2 Postfix 邮件接收流程

其基本的流程描述如下：

(1) 来自网络的邮件首先通过 smtpd 和 qmqpd 进程进入 Postfix 服务器，这两个进程去除了邮件中的 SMTP 或者 QMQP 协议封装，并对邮件进行初始的安全检查，以保护 Postfix 系统。

(2) 然后，这两个进程将发件人、收件人和消息内容传递给 cleanup 进程，用来按照规则拒绝不想要的邮件。

(3) cleanup 进程会对这些邮件进行最终的处理，包括加上丢失的 From 等信息头、转换邮件地址等工作。

(4) 最后，cleanup 进程把处理后的邮件作为单个文件放入 incoming 队列，并将新邮件到达的消息通知给该队列的管理进程。

另外，Postfix 还提供了与 Sendmail 兼容的命令，可以将本地的 Sendmail 邮件通过 postdrop 命令转送到 maildrop 队列，本地的 pickup 进程再从 maildrop 队列中将邮件读取出来，经过初步的安全检查后，将邮件传递给 cleanup 进程。即使在 Postfix 邮件系统没有运行的时候，这部分工作也能够正常进行。

trivial-rewrite 进程的作用是将邮件地址改写成标准的“用户名@邮件域名”的形式。

2. 邮件发送流程

邮件发送的流程如图 10-3 所示。

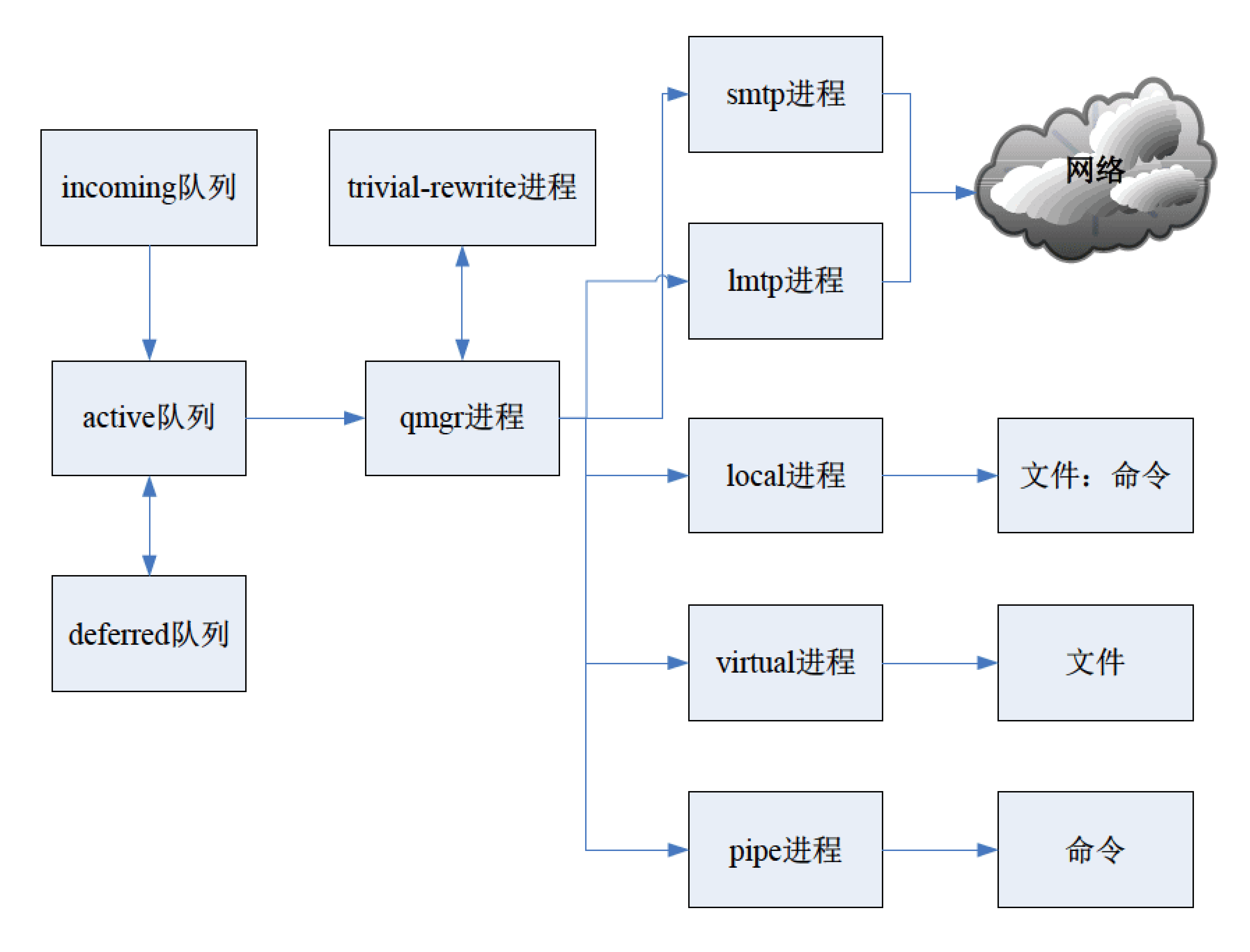


图 10-3　邮件发送流程

在发送邮件时，队列管理进程 qmgr 是整个系统的核心，它与 smtp、lmtp、local、virtual、pipe、discard 和 error 等邮件分发代理进程进行联系，要求它们根据收件人地址进行分发。discard 和 error 进程用于丢弃或者退回邮件。

active 队列保存了正在进行发送处理的邮件。而 deferred 队列保存着暂时不能分发的

邮件，这些邮件以后还会根据一定的策略进行重发操作。

trivial-rewrite 进程可以依照所定义的当地或者远程地址类分析每一个收件人的地址信息，还可以根据传输表加入相关的路由信息，以及查询 relocated 表确定收件人地址已经发生改变的邮件，这样的邮件会退回给发件人。

smtp 进程则根据目标主机寻找一个邮件接收服务器的列表，并按照一定的规则进行排序，再逐一与这些邮件服务器进行连接测试，直到收到响应。然后将发件人、收件人和邮件内容通过 SMTP 协议封装起来。

lmtp 进程的功能与 smtp 非常类似，但是采用的是 LMTP(Local Mail Transfer Protocol，本地邮件传输协议)，它是 SMTP 的升级版本。

local 进程的功能是邮件分发代理，它能够识别各种各样的邮件格式，并对这些邮件进行分发。多个 local 进程可以同时分发。通过配置，local 进程可以进行多种分发方式，包括：

- 将邮件分发到用户的邮箱。
- 将邮件分发到 procmail 等邮件客户端。
- 将邮件分发到 Postfix 分发代理进程。

virtual 进程的分发代理只负责分发到 UNIX 类的邮箱和 Qmail 邮件目录。它可以为各个子域分发邮件，因此特别适合企业内的邮件发送。

pipe 进程提供与其他邮件系统的外部接口，它通过管道给其他的命令提供邮件内容，并得到回应码。

3. master 进程

除了上面提到的这些进程外，Postfix 系统还运行着其他一些进程，这些进程都有特定的功能，还提供了 Postfix 命令的接口。

其中最重要的一个进程就是 master 进程。它监控着整个邮件系统中其他进程的工作，以 root 用户身份运行。master 进程与 Postfix 系统一起启动，一直运行到整个系统退出。所有其他的 Postfix 进程都是由 master 进程启动的，以 Postfix 用户的身份运行。

10.2.2 Postfix 服务器的安装与运行

安装 Postfix 服务器软件同样有两种方法：一种是通过 RPM 包的形式安装；另一种是以编译源码的形式安装。使用编译源码的形式安装比较复杂，不但要下载源码进行编译，还需要添加 Postfix 所需要使用的用户和用户组，之后还要对安装脚本进行修改，在此我们不介绍这种安装方法，有兴趣的读者可以参考其他资料。

使用 RPM 包的形式安装比较简单，在 CentOS 5.5 的安装光盘中已经附带了 Postfix 的 RPM 安装包，文件名为 postfix-2.3.3-2.1.el5_2.i386.rpm，我们只需要挂载光盘，然后使用 rpm 命令安装即可。默认情况下 CentOS 已将 Sendmail 安装到系统中了，如果要使用 Postfix，就必须先将 Sendmail 服务停止，然后再安装 Postfix 服务软件。具体的操作步骤如下。

(1) 关闭 Sendmail 服务。用下面的命令检查 Sendmail 服务是否已启动。

```
netstat -nutlp | grep :25
[root@CentOS ~]# netstat -nutlp | grep :25
tcp        0      0 127.0.0.1:25          0.0.0.0:*          LISTEN
4000/sendmail: acce
```

从上面的内容我们可以看出，Sendmail 服务正在监听 TCP 的 25 号端口。使用以下命令停止 Sendmail 服务以及取消 Sendmail 开机自动启动。

```
[root@CentOS ~]# /etc/rc.d/init.d/sendmail stop
Shutting down sm-client:                                [  OK  ]
Shutting down sendmail:                                  [  OK  ]
[root@CentOS ~]# chkconfig sendmail off
```

关闭自启动后，还可以用下面的命令来查看 Sendmail 的当前状态。

```
[root@CentOS ~]# chkconfig sendmail --list
sendmail       0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

(2) 安装 postfix 服务。将 CentOS 安装光盘挂载到系统中。

```
[root@CentOS /]# mount /dev/cdrom /mnt/cdrom
mount: block device /dev/cdrom is write-protected, mounting read-only
```

进入挂载的光盘目录，在 CentOS 目录中已经提供了 Postfix 的 RPM 安装包 postfix-2.3.3-2.1.el5_2.i386.rpm，运行以下命令即可完成安装。

```
 [root@CentOS CentOS]# rpm -ivh postfix-2.3.3-2.1.el5_2.i386.rpm
Preparing...            ########################################### [100%]
   1:postfix            ########################################### [100%]
```

安装完成后，所有的 Postfix 都处于默认值的状态，还无法进行实际的使用，需要进一步的配置。在下节我们将详细介绍如何配置 Postfix 服务。

(3) 启动 Postfix 服务。运行以下命令即可启动 Postfix 的服务进程。

```
[root@CentOS /]# /etc/init.d/postfix start
Starting postfix:                                     [  OK  ]
```

上述命令执行完成后，我们可以查看一下相关进程的情况来确认启动是否正常。

```
[root@CentOS /]# ps -eaf | grep postfix
root       6551     1  0 21:32 ?        00:00:00 /usr/libexec/postfix/master
postfix    6552  6551  0 21:32 ?        00:00:00 pickup -l -t fifo -u
postfix    6553  6551  0 21:32 ?        00:00:00 qmgr -l -t fifo -u
root       6583  4504  0 21:34 pts/0    00:00:00 grep postfix
```

可以看到，初始状态 Postfix 启动了 3 个进程。其中，主进程 master 是以 root 用户身份运行的，其他两个进程以 postfix 用户身份运行。下面我们再查看一下 25 端口，该端口是邮件服务器之间传递邮件的默认端口，也是客户端发送邮件时与服务器进行连接的默认端口。

```
[root@CentOS /]# netstat -anp | grep :25
tcp        0      0 127.0.0.1:25             0.0.0.0:*
LISTEN      6551/master
```

(4) 为了使 Postfix 服务器能够接受远程客户端的连接，还需要开放防火墙的对应端口。

```
[root@CentOS /]# iptables -I INPUT -p tcp --dport 25 -j ACCEPT
```

完成以上步骤后，虽然客户端已经可以通过 25 端口与 Postfix 服务器进行连接，但此

时 Postfix 还不能正常地收发邮件，需要进行进一步配置。

10.3 Postfix 服务器的配置

Postfix 服务器的配置相当复杂，除了 Postfix 软件外，还需要涉及很多其他的软件和知识，例如操作系统的用户认证、用户特权、数据库、DNS 配置等。本节将从最基本的 Postfix 配置讲起，再逐渐深入地介绍邮件的接收和发送、SMTP 认证等内容。

10.3.1 Postfix 服务器的基本配置

与 Sendmail 相比，Postfix 最被人称道的地方就在于其配置文件的可读性很高。Postfix 的主配置文件是/etc/postfix/main.cf。虽然该配置文件的内容比较多，但其中大部分内容都是注释，真正需要自行定义的选项并不多，而且这些选项就算不去定义，按照默认值也可以运行，只不过它只监听 127.0.0.1 这个接口的邮件收发。如果要使它能够支持客户端完成最基本的邮件收发任务，通常还需要进行下面的设置。

1. myhostname 和 mydomain 选项

用 myhostname 选项可以指定运行 Postfix 服务邮件主机的主机名称(FQDN 名)，用 mydomain 选项指定该主机的域名称。当然这两个选项也可以不进行设置，默认情况下，myhostname 选项被设置为本地主机名，而且 Postfix 会自动将 myhostname 选项值的第一部分删除并将其余部分作为 mydomain 选项的值，典型的设置如下所示：

```
myhostname = mail.test.edu
mydomain = test.edu
```

2. myorigin 选项

myorigin 选项实际上是设置由本台邮件主机寄出的每封邮件的邮件头中 mail from 的地址。由于 Postfix 默认使用本地主机名作为 myorigin 选项的值，因此一封由本地邮件主机寄出的邮件的邮件头中就会含有如“From: ‘snc’<snc@mail.test.edu>”这样的内容，它表明这封邮件是从 mail.test.edu 主机发来的。如果这台邮件服务器掌管着由多台及其主机组成的域时，应该将 myorigin 选项设置为本地邮件主机的域名(即“myorigin = test.edu 或$mydomain”)，这样一封由本地邮件主机寄出的邮件的邮件头中就会含有如“From: ‘snc’<snc@ test.edu >”这样的内容，显然更具有可读性。

```
Myori gin = $mydomain
```

3. inet_interfaces 选项

默认情况下，inet_interfaces 选项的值被设置为 localhost，这表明只能在本地邮件主机上寄信。如果邮件主机上有多个网络接口，而又不想使全部的网络接口都开放 Postfix 服务，就可以用主机名指定需要开放的网络接口。不过，通常是将所有的网络接口都开放，以便接收从任何网络接口来的邮件，即将 inet_interfaces 选项的值设置为 all。

```
inet_interfaces = all
```

4. mydestination 选项

mydestination 选项非常重要，它指定发往哪些域的邮件将会分发给本地用户，即只有当发来的邮件的收件人地址与该选项值相匹配时，Postfix 才会将该邮件接收下来，并传递给 local_transport 选项指定的分发代理，再由分发代理根据/etc/passwd 或/etc/aliases 等文件寻找收件人。例如，这里将该选项值设置为$mydomain 和$myhostname，表明无论来信的收件人地址是 xxx@test.edu(其中 xxx 表示某用户的邮件账户名)，还是 xxx@mail.test.edu，Postfix 都会接收这些邮件。

```
mydestination = $mydomain, $myhostname
```

5. mynetworks 和 mynetworks_style 选项

默认情况下，Postfix 将转发从授权网络范围的客户端到任何目的地的邮件。授权网络的范围可以使用 mynetworks 选项来设置。可将该选项值设置为所信任的某台主机的 IP 地址，也可设置为所信任的某个 IP 子网或多个 IP 子网(用“，”或者“ ”分隔)。这里，将 mynetworks 选项值设置为 192.168.44.0/24，则表示这台邮件主机只转发子网 192.168.111.0/24 中客户端所发来的邮件，而拒绝为其他子网转发邮件。

```
mynetworks = 192.168.111.0/24
```

除了 mynetworks 选项外，还有一个用于控制网络邮件转发的选项是 mynetworks_style，它主要用来设置可转发邮件网络的方式。通常有以下 3 种方式。

- class：在这种方式下，Postfix 会自动根据邮件主机的 IP 地址得知它所在的 IP 网络类型(即 A 类、B 类或是 C 类)，从而开放它所在的 IP 网段。例如，如果邮件主机的 IP 地址为 168.100.192.10，这是一个 B 类网络的 IP 地址，则 Postfix 会自动开放 168.100.0.0/16 整个 IP 网络。
- subnet：这是 Postfix 的默认值。Postfix 会根据邮件主机的网络接口上所设置的 IP 地址、子网掩码来得知所要开放的 IP 网段。例如，邮件主机的 IP 地址为 192.168.16.177，子网掩码为 255.255.255.192，则 Postfix 会开放 192.168.16.128/30 子网。
- host：在这种方式下，Postfix 只会开放本机。

通常，用户不设置 mynetworks-style 选项，而直接设置 mynetworks 选项。如果这两个参数都进行设置，那么只有 mynetworks 选项的设置有效。

6. relay_domains 选项

mynetworks 选项是针对邮件来源的 IP 来设置的，而 relay_domains 选项则是针对邮件来源的域名或主机名来设置的。例如，将该选项值设置为 test.edu，则表示任何由域 test.edu 发来的邮件都会被认为是可信任的，Postfix 会自动对这些邮件进行转发。

```
relay_domains = test.edu
```

完成了上面的基本设置后，重新启动 Postfix 服务，这台 Postfix 邮件主机就基本准备好了。但是目前它仅支持客户端发信，还不支持收信。下面来看一下这些选项配置的实例。

```
myhostname = mail.test.edu
mydomain = test.edu
myorigin = $myhostname      #本地发送邮件时，发件人主机设为mail.test.edu

#发往mail.test.edu、localhost.test.edu和localhost的邮件认为是发送给本地域
mydestination = localhost.$mydomain, $myhostname,localhost

mynetworks_style = subnet    #授权网络为Postfix服务器所在的子网。

#授权网络以外的客户端使用Postfix转发邮件时，其目的主机只能是mydestination指定的
域或者163.com域
relay_domains = & mydestination
relay_domains = 163.com
```

此外，要使它能在单位内部网络中更好地转发邮件，还必须进行 DNS 设置。本例中，可以在内部网络的 DNS 服务器上定义一个主区域 test.edu，并在该区域配置文件中定义了以下记录：

```
C:\Users\Administrator>telnet 192.168.111.133 25
mail.test.edu ESMTP Postfix
EHLO 192.168.111.134
250-mail.test.edu
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM:<1@test.edu>
250 2.1.0 Ok
RCPT TO:<2@test.edu>
250 2.1.5 Ok
RCPT TO:<mailtest@mail.test.edu>
DATA
354 End data with <CR><LF>.<CR><LF>
this is a mail.
.
250 2.0.0 Ok: queued as 956B6364A15
MAIL FROM:<1@test.edu>
250 2.1.0 Ok
RCPT TO:<2@163.com>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
this is another mail.
.
250 2.0.0 Ok: queued as C1399364A15
quit
221 2.0.0 Bye
```

从上面的测试内容可以看出，现在 Postfix 已经能够正常地发送邮件了。下面我们再以 mailtest 用户登录到服务器，查看刚才是否收到邮件。

```
CentOS release 5.5 (Final)
Kermel 2.6.10-194.el5 on an i686
```

```
CentOS login: mailtest             #使用 mailtest 用户登录
Password:
Last login: Thu Mar  1 21:14:32 on tty1
[mailtest@CentOS ~]$ mail          #查看收到的邮件
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/mailtest": 1 message 1 new
>N  1 2@test.edu         Thu Mar  1 21:13  14/482    #收到的邮件
```

其中，mail 命令是 Linux 系统中邮件客户端的启动命令，但只是一个简单字符界面的邮件客户端。在实际使用中，我们很少使用它。

10.3.2　配置 Postfix 接收域

作为邮件接收服务器，Postfix 不但可以设置接收本地邮件或者属于其域名中的邮件，也可以配置成接收许多其他类型域的最终目的地。这些域和 Postfix 服务器的主机名没有直接的联系，这些域名一般称为托管域。

设置托管域的方式很简单，只需要在 main.cf 中的 mydestination 选项中将所需要的域名添加到其列表中，那么 Postfix 就会开始接收此域的所有邮件。例如下面的例子。

```
mydestination = localhost.$mydomain, $myhostname,localhost, test2.edu
```

从例子中可以看出，这是在 main.cf 基本配置文件中的 mydestination 选项中新添加了一个参数“test2.edu”，那么 Postfix 就会接收发往“test2.edu”这个域的所有邮件。但是，这种设置托管域的方法并不好，除了设置此功能的管理员，其他人很难从此选项中看出哪一个是本地域，哪一个是托管域。为了解决这个问题，我们可以使用虚拟别名域，将发给虚拟域的邮件实际投递到真实域的用户邮箱中；可以实现群组邮递的功能，即指定一个虚拟邮件地址，任何人发给这个邮件地址的邮件都将由邮件服务器自动转发到真实域中一组用户的邮箱中。

这里的虚拟域可以是实际并不存在的域，而真实域既可以是本地域(即 main.cf 文件中 mydestination 参数值中列出的域)，也可以是远程域或 Internet 中的域。虚拟域是真实域的一个别名。实际上，通过一个虚拟别名表(Virtual)，实现了虚拟域的邮件地址到真实域邮件地址的重定向。

下面通过一个实例来说明虚拟别名域的设置方法。

如果要将发送给虚拟域@VirtualDomian.com 的邮件实际投递到真实的本地域@RealDomian.com，那么可在虚拟别名表中进行如下定义：

```
@VirtualDomian.com  @RealDomian.com
```

如果要将发送给虚拟域的某个虚拟用户(或组)的邮件实际投递到本地 Linux 系统中某个用户账户的邮箱中，那么可在虚拟别名表中进行如下定义：

```
VirtualUser@VirtualDomian.com  RealUser
VirtualGroup@VirtualDomian.com  RealUser1, RealUser2, RealUser3
```

如果要将发送给虚拟域中的某个虚拟用户(或组)的邮件实际投递到本地 Linux 系统中和 Internet 中某个用户账户的邮箱中，那么可在虚拟别名表中进行如下定义：

```
VirtualUser@VirtualDomian.com  VirtualUser, User@RealDomian.com
```

在实际应用中，配置虚拟别名域，必须按以下步骤进行。

(1) 编辑 Postfix 主配置文件/etc/postfix/main.cf，进行如下定义：

```
virtual_alias_domains = VirtualDomain.com，RealDomain.com
virtual_alias_maps = hash:/etc/postfix/virtual
```

其中，virtual_alias_domains 选项用来指定虚拟别名域的名称，virtual_alias_maps 选项用来指定含有虚拟别名域定义的文件路径。

(2) 编辑配置文件/etc/postfix/virtual，进行如下定义：

```
@VirtualDomian.com  @RealDomian.com
VirtualUser@VirtualDomian.com  RealUser
VirtualGroup@VirtualDomian.com  RealUser1，RealUser2，RealUser3
VirtualUser@VirtualDomian.com  VirtualUser，User@RealDomian.com
```

(3) 在修改配置文件 main.cf 和 virtual 后，要使更改立即生效，应分别执行/usr/sbin 目录下的两条命令。

```
[root@CentOS ~]# postmap /etc/postfix/virtualpostfix reload
[root@CentOS ~]# postfix reload
postfix/postfix-script: refreshing the Postfix mail system
```

其中，第 1 条命令用来将文件/etc/postfix/virtual 生成 Postfix 可以读取的数据库文件/etc/postfix/virtual.db；第 2 条命令用于重新加载 Postfix 主配置文件 main.cf 文件。

另外，Postfix 还可以作为其他域的后备邮件网关服务器。在通常情况下，Postfix 不会接收主邮件服务器所在域的邮件，但当主邮件服务器发生故障的时候，后备邮件网关就可以临时接收此域的所有邮件，当主邮件服务器重新开始工作的时候，Postfix 又会将所有接收到的邮件转发给主邮件服务器，这些域可以在中继域地址类中定义。

10.3.3 配置 SMTP 认证

细心的读者可以发现，前面我们配置的 Postfix 服务器没有认证机制，任何一个客户端都可以通过 SMTP 与 Postfix 服务器进行连接，然后使用 RCPT 命令要求 Postfix 服务器转发邮件到收件人所在的邮件服务器，也就是说，互联网上的任何计算机，不需要使用账号就可以通过这台邮件服务器向任何邮箱发送邮件，这无疑为垃圾邮件的发送敞开了大门。

为了解决这个问题，我们需要在 SMTP 服务器中使用身份认证机制，只有通过身份认证的用户才能发送 SMTP 请求服务器发送邮件到目的地。认证账号一般与接收邮件的账号相同，这些账号可以是操作系统账号，也可以是虚拟账号，或者是保存在数据库中的用户账号。

目前，比较常见的 SMTP 认证机制是通过 Cyrus SASL 软件包来实现的。Cyrus SASL 是 Cyrus Simple Authentication and Security Layer 的简写，它最大的功能是为应用程序提供了认证函数库。应用程序可以通过函数库所提供的功能定义认证方式，并让 SASL 通过与邮件服务器主机的沟通从而提供认证的功能。

下面介绍使用 Cyrus SASL 包实现 SMTP 认证的具体方法。

(1) Cyrus-SASL 认证包的安装。

使用下面的命令检查系统是否已经安装了 Cyrus-SASL 认证包或查看已经安装了何种

版本。

```
[root@CentOS /]# rpm -qa | grep sasl
cyrus-sasl-plain-2.1.22-5.el5_4.3
cyrus-sasl-lib-2.1.22-5.el5_4.3
cyrus-sasl-md5-2.1.22-5.el5_4.3
cyrus-sasl-2.1.22-5.el5_4.3
```

可以看到，早 CentOS 5.5 中已经默认安装了 SASL 的相关组件。如果使用上面的命令没有显示 cyrus-sasl-2.1.22 安装包，可以使用以下命令将安装光盘中的 cyrus-sasl-2.1.22-5.el5_4.3.i386.rpm 安装包安装到系统中。

```
[root@CentOS /]# rpm -ivh cyrus-sasl-2.1.22-5.el5_4.3.i386
```

(2) Cyrus-SASL V2 的密码验证机制。

安装完成后，主要复制的文件包括/usr/sbin 目录中的 saslauthd，它负责提供安全认证功能。默认情况下，Cyrus-SASL V2 版使用 saslauthd 这个守护进程进行密码认证，而密码认证的方法有多种，使用下面的命令可查看当前系统中的 Cyrus-SASL V2 所支持的密码验证机制。

```
[root@CentOS ~]# saslauthd -v
saslauthd 2.1.22
authentication mechanisms: getpwent kerberos5 pam rimap shadow ldap
```

从上面显示的内容可以看到，当前可使用的密码验证方法有 getwent、kerberos5、pam、rimap、shadow 和 ldap。为简单起见，这里准备采用 shadow 验证方法，也就是直接用 /etc/shadow 文件中的用户账户及密码进行验证。因此，在配置文件 /etc/sysconfig/saslauthd 中，应修改当前系统所采用的密码验证机制为 shadow，即：

```
MECH=shadow
```

修改完成后，就可以使用以下命令启动认证。

```
[root@CentOS /]# /usr/sbin/saslauthd -m /var/run/saslauthd/ -a shadow
                    #-a shadow 意味着使用 shadow 认证方式
[root@CentOS /]# ps -eaf | grep sasl
root      5013     1  0 10:49 ?        00:00:00 /usr/sbin/saslauthd -m
/var/run/saslauthd/ -a shadow
root      5014  5013  0 10:49 ?        00:00:00 /usr/sbin/saslauthd -m
/var/run/saslauthd/ -a shadow
root      5015  5013  0 10:49 ?        00:00:00 /usr/sbin/saslauthd -m
/var/run/saslauthd/ -a shadow
root      5016  5013  0 10:49 ?        00:00:00 /usr/sbin/saslauthd -m
/var/run/saslauthd/ -a shadow
root      5017  5013  0 10:49 ?        00:00:00 /usr/sbin/saslauthd -m
/var/run/saslauthd/ -a shadow
root      5034  4911  0 10:50 pts/1      00:00:00 grep sasl
```

(3) 测试 Cyrus-SASL V2 的认证功能。

通过查看进程，我们可以看到默认情况下启动了 5 个 saslauthd 进程。为了检验 SASL 安全认证是否已经正常工作，可以输入以下命令进行测试。

```
[root@CentOS /]# /usr/sbin/testsaslauthd -u root -p 111111
0: OK "Success."
```

testsaslauthd 文件是用来检验一个账号是否可以通过 SASL 安全认证。其中，“-u”是指定用户名，“-p”指定密码。

(4) 启用 smtp 认证。

默认情况下，Postfix 并没有启用 SMTP 认证机制。要让 Postfix 启用 SMTP 认证，就必须对 Postfix 的主配置文件/etc/postfix/main.cf 进行修改。其中主要的配置命令如下：

```
smtpd_sasl_auth_enable = yes
smtpd_sasl_local_domain = ''
smtpd_recipient_restrictions =
permit_mynetworks,permit_sasl_authenticated, reject_unauth_destination
broken_sasl_auth_clients=yes
smtpd_client_restrictions = permit_sasl_authenticated
smtpd_sasl_security_options = noanonymous
```

其中：

- smtpd_sasl_auth_enable：指定是否要启用 SASL 作为 SMTP 认证方式。默认不启用，这里必须启用它，所以要将该参数值设置为 yes。
- smtpd_sasl_local_domain：如果采用 Cyrus-SASL V2 版进行认证，那么这里不做设置。
- smtpd_recipient_restrictions：表示通过收件人地址对客户端发来的邮件进行过滤。通常有以下几种限制规则。
  - ◆ permit_mynetworks：表示只要是收件人地址位于 mynetworks 参数中指定的网段就可以转发邮件。
  - ◆ permit_sasl_authenticated：表示允许转发通过 SASL 认证的邮件。
  - ◆ reject_unauth_destination：表示拒绝转发含未信任目标地址的邮件。
  - ◆ broken_sasl_auth_clients：表示是否兼容非标准的 SMTP 认证。有一些 Microsoft 的 SMTP 客户端(如 Outlook Express 4.x)采用非标准的 SMTP 认证协议，只需将该参数设置为 yes 就可解决这类不兼容问题。
- smtpd_client_restrictions：表示限制可以向 Postfix 发起 SMTP 连接的客户端。如果要禁止未经过认证的客户端向 Postfix 发起 SMTP 连接，则可将该参数值设置为 permit_sasl_authenticated。
- smtpd_sasl_security_options：用来限制某些登录的方式。如果将该参数值设置为 noanonymous，则表示禁止采用匿名登录方式。

(5) 验证认证方式。

上述的配置完成后，在重新启动 Postfix 服务之前，首先检测一下 Postfix 是否已经支持 SASL 认证，使用以下的命令：

```
[root@CentOS postfix]# postconf -a
cyrus
dovecot
```

其中，postconf 命令用来显示 Postfix 当前的配置状态，“-a”选项表示输出当前支持的 SASL 认证类型，从上面的输出我们可以看到，现在 Postfix 已经支持 CYRUS 的 SASL 认证了。

(6) 重启服务。

在完成上述步骤后，必须使用/etc/init.d/postfix reload 命令重新载入配置文件，或使用

/etc/init.d/postfix restart 命令重新启动 Postfix 服务。

所有的工作完成后，我们可以对 Postfix 的 SMTP 认证进行测试。具体的测试代码如下：

```
[root@CentOS /]# telnet 192.168.111.133 25
Trying 192.168.111.133...
Connected to test.edu (192.168.111.133).
Escape character is '^]'.
220 mail.test.edu ESMTP Postfix
EHLO 192.168.111.134
250-mail.test.edu
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH PLAIN CRAM-MD5 DIGEST-MD5 LOGIN     #SMTP 认证
250-AUTH=PLAIN CRAM-MD5 DIGEST-MD5 LOGIN     #SMTP 认证
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

从上面的登录信息我们可以看到，与之前的登录信息相比，多出了两行的显示内容，表明现在 Postfix 服务器已经支持 SMTP 认证了。

10.4　架设 POP3 和 IMAP 服务器

前面我们已经详细介绍了 Postfix 服务器本身的配置，但 Postfix 只承担了邮件系统中的 MTA 功能，一个完整的邮件系统还需要很多其他的功能，例如，POP/IMAP 服务、Web 界面客户端、垃圾邮件过滤等。这些功能 Postfix 都是无法完成的，需要第三方软件的支持，从本节开始，我们将介绍这些软件的安装、运行和配置方法。

10.4.1　Dovecot 介绍

Dovecot 软件是一种在 Linux 下使用的开源软件，它可以提供 POP3 和 IMAP 服务。Dovecot 将安全作为主要的设计目标，而且速度快、占用内存小、配置简单，可以在各种规模的场合使用。

在实际使用中，常见的邮件系统是 Postfix、Dovecot 和 MySQL 三者配合搭建的邮件服务器。其中 Postfix 作为邮件发送服务，Dovecot 作为邮件接收服务，而 MySQL 作为账号存储服务器。

10.4.2　Dovecot 服务的安装

首先，我们来查看一下默认的 CentOS 是否已经安装了 Dovecot 相关的服务包，使用以下的命令：

```
[root@CentOS /]# rpm -q dovercot
package dovercot is not installed
```

可以看到，CentOS 5.5 的默认安装是没有安装 Dovecot 服务的，我们需要在安装光盘中找到需要的安装包，包括 dovecot-1.0.7-7.el5.i386.rpm、mysql-5.0.77-4.el5_4.2.i386.rpm

和 perl-DBI-1.52-2.el5.i386.rpm，然后逐个安装这些软件包，如下面的内容所示：

```
[root@CentOS usr]# rpm -ivh perl-DBI-1.52-2.el5.i386.rpm
Preparing...          ########################################### [100%]
      package perl-DBI-1.52-2.el5.i386 is already installed
[root@CentOS usr]# rpm -ivh mysql-5.0.77-4.el5_4.2.i386.rpm
Preparing...          ########################################### [100%]
      package mysql-5.0.77-4.el5_4.2.i386 is already installed
[root@CentOS usr]# rpm -ivh dovecot-1.0.7-7.el5.i386.rpm
Preparing...          ########################################### [100%]
      package dovecot-1.0.7-7.el5.i386 is already installed
```

10.4.3 Dovecot 服务的配置

Dovecot 服务的主配置文件位于/etc/dovecot.conf，其可配置内容非常丰富，并且每一个选项都有详细的说明。但这里我们只介绍基本的配置选项，对其他选项有兴趣的读者可以自行研究。

基本的配置内容如下：

```
protocols = imap pop3     #开启 POP3 和 IMAP 服务
ssl_disable = YES         #禁止安全连接
passdb passwd {            #使用/etc/passwd 认证文件
   }
passdb shadow {            #使用/etc/shadow 认证文件
   }
```

配置完成后，使用以下命令来启动 Dovecot 服务。

```
[root@CentOS /]# /etc/rc.d/init.d/dovecot start
Starting Dovecot Imap:                               [  OK  ]
[root@CentOS /]# chkconfig --level 345 dovecot on
[root@CentOS /]# ps -eaf | grep dovecot
root     11182     1  0 15:18 ?        00:00:00 /usr/sbin/dovecot
root     11184 11182  0 15:18 ?        00:00:00 dovecot-auth
dovecot  11185 11182  0 15:18 ?        00:00:00 pop3-login
dovecot  11186 11182  0 15:18 ?        00:00:00 pop3-login
dovecot  11187 11182  0 15:18 ?        00:00:00 pop3-login
dovecot  11188 11182  0 15:18 ?        00:00:00 imap-login
dovecot  11189 11182  0 15:18 ?        00:00:00 imap-login
dovecot  11190 11182  0 15:18 ?        00:00:00 imap-login
root     11206 11155  0 15:19 pts/0    00:00:00 grep dovecot
```

从上面的进程列表中我们可以看出，Dovecot 服务包含了两个使用 root 用户运行的进程以及 6 个使用 dovecot 用户运行的进程。其中，dovecot 用户是在安装 Dovecot 时自动创建的。

下面我们再来查看一下 POP3 和 IMAP 服务响应的端口是否已经处于监听状态，如下面的命令所示：

```
[root@CentOS /]# netstat -anp | grep :110
tcp        0      0 :::110                      :::*                        LISTEN
11182/dovecot
[root@CentOS /]# netstat -anp | grep :143
tcp        0      0 :::143                      :::*                        LISTEN
11182/dovecot
```

从显示内容可以看出，110 端口和 143 端口已经由 dovecot 进程进行监听，为了向其他用户提供服务，还需要在防火墙中添加开放端口的命令，如下所示：

```
[root@CentOS /]# iptables -I INPUT -p tcp --dport 110 -j ACCEPT
[root@CentOS /]# iptables -I INPUT -p tcp --dport 143 -j ACCEPT
```

完成以上的设置后，我们可以对 Dovecot 的 POP3 服务和 IMAP 服务进行简单的测试，内容如下：

```
[root@CentOS /]# telnet 192.168.111.133 110  #与服务器的 POP3 端口 110 连接
Trying 192.168.111.133...
Connected to test.edu (192.168.111.133).
Escape character is '^]'.
+OK Dovecot ready.
user mailtest                                  #输入用户名
+OK
pass 123456                                    #输入密码
+OK Logged in.
Stat                                           #列出邮箱的邮件数和字节数
+OK 1 449
list 1                                         #列出第一封邮件的字节数
+OK 1 449
retr 1                                         #读取第一封邮件的内容
+OK 449 octets
Return-Path: <test@abc.com>
X-Original-To: mailtest@mail.test.edu
Delivered-To: mailtest@mail.test.edu
Received: from 192.168.44.134 (unknown [192.168.44.133])
       by mail.test.edu (Postfix) with ESMTP id 46AFA364A14
       for <mailtest@mail.test.edu>; Thu,  1 Mar 2012 21:12:23 +0800 (CST)
Message-Id: <20120301131256.46AFA364A14@mail.test.edu>
Date: Thu,  1 Mar 2012 21:12:23 +0800 (CST)
From: test@abc.com
To: undisclosed-recipients:;

test.
.dele 1                                        #给第一封邮件打上删除标记
+OK Marked to be deleted.
stat                                           #再次查看邮箱，邮件数为 0
+OK 0 0
rset                                           #撤销所有删除标记
+OK
quit                                           #退出 POP3 服务器
+OK Logging out.
Connection closed by foreign host
[root@CentOS /]# telnet 192.168.111.133 143   #连接 IMAP 服务端口 143
Trying 192.168.111.133...
Connected to test.edu (192.168.111.133).
Escape character is '^]'.
* OK Dovecot ready.
A LOGIN mailtest 123456                        #使用 mailtest 登录
A OK Logged in.
A SELECT INBOX                                 #选择 INBOX 邮箱
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft \*)]
Flags permitted.
* 1 EXISTS
```

```
* 0 RECENT
* OK [UNSEEN 1] First unseen.
* OK [UIDVALIDITY 1330674271] UIDs valid
* OK [UIDNEXT 2] Predicted next UID
A OK [READ-WRITE] Select completed.
A FETCH 1 body[header]                            #提取第一封邮件的内容
* 1 FETCH (FLAGS (\Seen) BODY[HEADER] {442}
Return-Path: <test@abc.com>
X-Original-To: mailtest@mail.test.edu
Delivered-To: mailtest@mail.test.edu
Received: from 192.168.44.134 (unknown [192.168.44.133])
       by mail.test.edu (Postfix) with ESMTP id 46AFA364A14
       for <mailtest@mail.test.edu>; Thu,  1 Mar 2012 21:12:23 +0800
(CST)
Message-Id: <20120301131256.46AFA364A14@mail.test.edu>
Date: Thu,  1 Mar 2012 21:12:23 +0800 (CST)
From: test@abc.com
To: undisclosed-recipients:;

)
A OK Fetch completed.
A LOGOUT                                           #退出 IMAP 服务器
* BYE Logging out
A OK Logout completed.
Connection closed by foreign host.
```

从上面的测试内容可以看出，Dovecot 的 POP3 和 IMAP 服务已经能够正常地使用了。另外，当 Dovecot 与 Postfix 集成使用时，最重要的是配置相关的认证方式和邮箱的位置，这些配置都可以在/etc/dovecot.conf 中通过相应的选项进行设置，读者可以自行研究。

10.5 基于 Web 方式的邮件服务器配置

在上节中，我们已经实现了一个以 POP3 和 IMAP 为协议的 E-Mail 邮件服务器，并且使用字符界面的形式成功地收发了邮件。但是，在实际使用中，使用字符界面收发邮件是非常不方便的。在本节中，我们将介绍如何使用 Squirrelmail 软件来架设一台支持 Web 界面访问的邮件服务器。

10.5.1 Squirrelmail 介绍

除了使用 Outlook、Foxmail 等邮件客户端来收发电子邮件外，还有一种流行的方式是使用 Web 界面来收发邮件，它的优点是只要客户机中装有 Web 浏览器即可，不需要安装其他软件。为了能够让用户使用 Web 界面访问邮箱，首先要架设 Web 服务器，然后需要采用 Web 语言编写 Web 程序，Web 程序再与邮件服务器进行交互，以帮助用户使用 Web 界面的方式来收发邮件。

Squirrelmail(中文名为“小松鼠网页电子邮件系统”)便是这样一款 Web 程序，它能够与 Postfix 集成在一起，为用户提供 Web 邮件服务。它是一款使用 PHP4 编写、基于 IMAP 协议的 Webmail 电子邮件客户端软件。Squirrelmail 的主要特点是架设、操作与维护简便，无须使用 SQL Server，与标准的电子邮件服务软件(如 Postfix)的兼容性好，而且

可以通过安装插件(Plugin)来扩充其功能。当需要更新版本或使用新的插件时，可以直接到 Squirrelmail 官方网站 http://www.squirrelmail.org/上下载。

10.5.2　Squirrelmail 的安装

读者可使用下面的命令检查系统是否已经安装了 Squirrelmail。

```
[root@CentOS ~]#  rpm -q squirrelmail
package squirrelmail is not installed
```

可以看出，在 CentOS 操作系统中，默认并没有安装 Squirrelmail，用户需要安装 Squirrelmail 软件及 PHP 支持包，文件名分别为：squirrelmail-1.4.8-5.el5.centos.10.noarch.rpm 和 php-mbstring-5.1.6-27.el5.i386.rpm。安装过程如下面的代码所示。

```
[root@CentOS usr]# rpm -ivh php-mbstring-5.1.6-27.el5.i386.rpm
Preparing...             ########################################### [100%]
   1:php-mbstring        ########################################### [100%]
[root@CentOS usr]# rpm -ivh squirrelmail-1.4.8-
5.el5.centos.10.noarch.rpm
Preparing...             ########################################### [100%]
   1:squirrelmail        ########################################### [100%]
```

10.5.3　Squirrelmail 的配置

Squirrelmail 的主配置文件为/etc/squirrelmail/config.php。要配置 Squirrelmail，可以直接修改该文件的内容，但是使用 Squirrelmail 的配置工具来配置更方便、更直观。在/usr/share/quirrelmail/config/目录中的文件 conf.pl 为对应的配置文件，它是使用 pl 语言编写的程序。使用 squirrelmail 的配置工具进行配置的具体步骤如下。

(1) 打开 squirrelmail 的配置工具，可执行下面的命令。

```
[root@CentOS ~]# perl /usr/share/squirrelmail/config/conf.pl
```

命令执行后打开设置主菜单，如图 10-4 所示。

```
SquirrelMail Configuration : Read: config.php (1.4.0)
---------------------------------------------------------
Main Menu --
1.  Organization Preferences
2.  Server Settings
3.  Folder Defaults
4.  General Options
5.  Themes
6.  Address Books
7.  Message of the Day (MOTD)
8.  Plugins
9.  Database
10. Languages

D.  Set pre-defined settings for specific IMAP servers

C   Turn color off
S   Save data
Q   Quit

Command >> d_
```

图 10-4　进入配置页面

(2) 选择主菜单项 D，即在命令提示符(Command>>)后输入字母“d”，则进入指定 IMAP 服务器的预设置，如图 10-5 所示。然后可根据 Linux 系统的当前配置，在命令提示

符后输入所用的 IMAP 服务器类型名，此处输入“dovecot”。

```
SquirrelMail Configuration : Read: config.php
---------------------------------------------------------
While we have been building SquirrelMail, we have discovered some
preferences that work better with some servers that don't work so
well with others.  If you select your IMAP server, this option will
set some pre-defined settings for that server.

Please note that you will still need to go through and make sure
everything is correct.  This does not change everything.  There are
only a few settings that this will change.

Please select your IMAP server:
    bincimap    = Binc IMAP server
    courier     = Courier IMAP server
    cyrus       = Cyrus IMAP server
    dovecot     = Dovecot Secure IMAP server
    exchange    = Microsoft Exchange IMAP server
    hmailserver = hMailServer
    macosx      = Mac OS X Mailserver
    mercury32   = Mercury/32
    uw          = University of Washington's IMAP server

    quit        = Do not change anything
Command >> dovecot
```

图 10-5　配置邮件服务器类型

(3) 输入 IMAP 服务器的配型后，程序会自动检测此服务器中是否已经运行 Dovecot 服务，如果检测到此服务，将自动显示服务的相关内容，如图 10-6 所示。

```
Command >> dovecot

              imap_server_type = dovecot
           default_folder_prefix = <none>
                   trash_folder = Trash
                    sent_folder = Sent
                   draft_folder = Drafts
             show_prefix_option = false
           default_sub_of_inbox = false
show_contain_subfolders_option = false
             optional_delimiter = detect
                  delete_folder = false

Press any key to continue..._
```

图 10-6　检测到的服务器信息

(4) 选择主菜单项 1，进入组织设置子菜单，如图 10-7 所示。然后可根据具体情况修改组织的名称、标志等信息。设置完毕后在命令提示符后输入字母“r”，即可返回主菜单。

```
SquirrelMail Configuration : Read: config.php (1.4.0)
---------------------------------------------------------
Organization Preferences
1.  Organization Name      : SquirrelMail
2.  Organization Logo      : ../images/sm_logo.png
3.  Org. Logo Width/Height : (308/111)
4.  Organization Title     : SquirrelMail $version
5.  Signout Page           :
6.  Top Frame              : _top
7.  Provider link          : http://www.squirrelmail.org/
8.  Provider name          : SquirrelMail

R   Return to Main Menu
C   Turn color off
S   Save data
Q   Quit

Command >> _
```

图 10-7　组织信息子菜单

(5) 选择主菜单项 2，进入服务器设置子菜单，如图 10-8 所示。由于前面已经对 IMAP 服务器做了预设置，因此这里只需要将服务器的域名(子菜单项 1)修改为

“test.edu”，将发送邮件的方式(子菜单项 3)改为“SMTP”(此时更新 SMTP 设置项即子菜单项 B 也随之变为“localhost:25”)。

```
SquirrelMail Configuration : Read: config.php (1.4.0)
---------------------------------------------------------
Server Settings

General
-------
1.  Domain                 : test.edu
2.  Invert Time            : false
3.  Sendmail or SMTP       : SMTP

A.  Update IMAP Settings   : localhost:143 (dovecot)
B.  Update SMTP Settings   : localhost:25

R   Return to Main Menu
C   Turn color off
S   Save data
Q   Quit

Command >> _
```

图 10-8　服务器设置子菜单

(6) 选择主菜单项 4，进入全局设置子菜单，如图 10-9 所示。在这里可以对服务器的各种全局内容进行详细的设置。

```
3.  Directory Hash Level          : 0
4.  Default Left Size             : 150
5.  Usernames in Lowercase        : true
6.  Allow use of priority         : true
7.  Hide SM attributions          : false
8.  Allow use of receipts         : true
9.  Allow editing of identity     : true
    Allow editing of name         : true
    Remove username from header   : false
10. Allow server thread sort      : true
11. Allow server-side sorting     : true
12. Allow server charset search   : true
13. Enable UID support            : true
14. PHP session name              : SQMSESSID
15. Location base                 :
16. Only secure cookies if poss.  : true
17. Disable secure forms          : false
18. Page referal requirement      :

R   Return to Main Menu
C   Turn color off
S   Save data
Q   Quit

Command >> _
```

图 10-9　全局设置子菜单

(7) 选择主菜单项 10。进入语言设置子菜单，如图 10-10 所示。为了使 Web 页面支持中文，需要将默认语言(子菜单项 1)改为“zh_CN”(中文)，将默认字符集(子菜单项 2)改为“gb2312”。

(8) 选择主菜单项 S，即可将所做的修改同时保存在文件/etc/squirrelmail/config.php 和 usr/share/squirrelmail/config/config.php(符号连接文件)中，如图 10-11 所示。

上述修改只是对 Squirrelmail 的最基本配置，还可根据需要做进一步的修改。特别是，在默认情况下仅配置安装了 3 个插件，而 Squirrelmail 的插件非常丰富，因此需要时可选择主菜单项 8，将其他插件配置到 Squirrelmail 中。

```
SquirrelMail Configuration : Read: config.php (1.4.0)
---------------------------------------------------------
Language preferences
1.  Default Language         : zh_CN
2.  Default Charset          : gb2312
3.  Enable lossy encoding    : false

R   Return to Main Menu
C   Turn color off
S   Save data
Q   Quit

Command >> _
```

图 10-10 语言设置子菜单

```
Command >> s

Data saved in config.php
Press enter to continue..._
```

图 10-11 保存设置

在完成了对 Squirrelmail 的配置后，为了能在浏览器中访问到 Squirrelmail 的 Webmail 网页，应执行下面的命令来重新启动 httpd 服务。

```
[root@CentOS /]# /etc/init.d/httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd:                                            [  OK  ]
```

10.5.4 Squirrelmail 测试

因此，可直接在浏览器的地址栏中输入“http://mail.test.edu/webmail”(其中，mail.test.edu 是 postfix+apache+squirrelmail 服务器的域名)，从而打开 Squirrelmail 的登录页面，如图 10-12 所示。

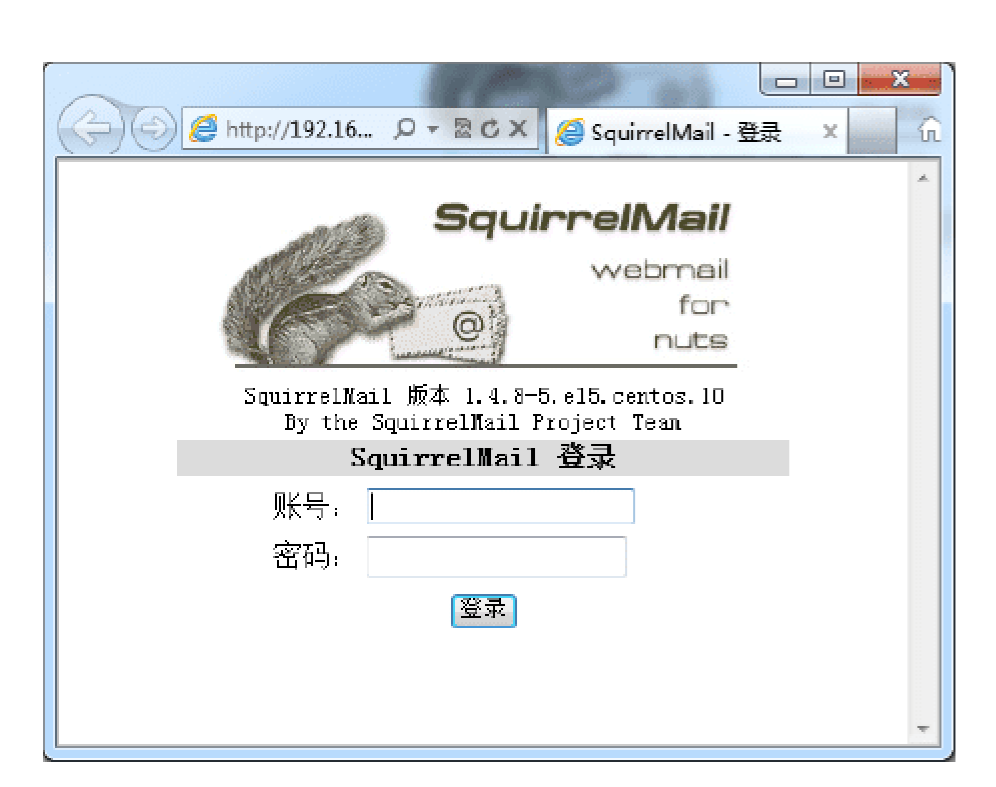


图 10-12 Squirrelmail 登录页面

在 Squirrelmail 的登录页面中，输入 Linux 系统中的用户名及密码后，系统即可进入 Webmail 中。接下来，用户就可以正常地收发邮件了。

Squirrelmail 需要和其他软件配合使用才能正常工作，包括支持 PHP4 的 Apache 服务器、Postifx 服务器、用 dovecot 安装的 IMAP 服务器等。因此，如果运行时出现问题，除了从 Squirrelmail 自身寻找问题外，还需要注意其他服务是否配置正确。

10.6　Mail 服务的邮件过滤功能

自从电子邮件诞生的那一天起，垃圾邮件就如影随形地一直伴随着其发展。垃圾邮件不但经常被用来作为电脑病毒的传播手段，它还浪费了大量的网络带宽和系统资源。

10.6.1　Procmail 介绍

在 Linux 系统中，最常见的是使用 Sendmail、Postfix 或者 Qmail 等软件作为 MTA，再配合 POP3/IMAP 服务器来组成一个基本的邮件系统。这样的系统可以很好地满足收发邮件的要求，但这样的系统对付垃圾邮件是无能为力的，虽然 Postfix 自带了黑白名单等简单的邮件过滤功能，但由于其规则简单，过滤垃圾邮件的效果和功能都很一般。因此，我们还是需要第三方软件来承担反垃圾邮件的任务。

在 Linux 平台下的开源软件中，Procmail 是一个非常不错的选择。Procmail 是一个可以自定义的强大的邮件过滤工具。系统管理员可以通过在客户端或者服务器配置 Procmail 来过滤垃圾邮件。

10.6.2　Procmail 的安装

在 CentOS 5.5 中，默认情况下是安装 Procmail 的，管理员可以使用以下命令来查看系统中是否已经安装 Procmail。

```
[root@CentOS /]# rpm -qa | grep procmail
procmail-3.22-17.1.el5.centos
```

从命令显示内容中我们可以看出，Procmail 3.22 版已经安装在系统中，如果系统中没有安装，管理员也可以在 CentOS 的安装盘中找到 RPM 安装文件进行安装，RPM 软件安装方法我们前面已经多次介绍过，在此不再累述。还有一种软件的获取途径是访问 Procmail 的官方网站 http://www.procmail.org/来获得程序的源代码进行编译安装。

10.6.3　Procmail 的配置

安装完成后，还需要对 Procmail 进行配置才能使其生效。Procmail 中有两种配置文件，一种是作用于所有邮件用户的配置文件，位于/etc/procmailrc 文件；另一种是只针对某个邮件用户有效的配置文件，位于每一个用户的主目录下的 procmailrc 文件，此文件只过滤该用户的邮件，对其他用户不起作用。

实际上，Procmail 安装完成后，.etc.procmailrc 文件并不存在，需要用户手动建立该文件，此文件中应该按照格式书写若干的过滤规则，Procmail 会根据这些规则进行邮件过滤，规则的具体格式如下：

```
:0 [flags] [:[localhostfile]]  #设置被过滤邮件检测的位置
[conditions]                   #设置被检测的规则内容，可多行，每行一个内容
[action]                       #设置对符合规则内容的操作，可多行，每行一个操作
```

其中，“:0”表示一个新的规则开始，flags 是作为位置的标识，可用的参数如表 10-5 所示。

表 10-5　flags 的可用参数

参　数	说　明
H	对邮件的头部进行检查
B	对邮件的正文部分进行检查
h	将邮件的头部数据导入管道(pipe)、文件或者其他邮件并导向到在规则中指定的地方
b	将邮件的正文数据导入管道(pipe)、文件或者其他邮件并导向到在规则中指定的地方
D	区分字母大小写

在 conditions 部分可以使用多行，每行以“*”开头，其中常用的符号如表 10-6 所示。

表 10-6　conditions 的常用符号及其说明

符　号	说　明
!	反向选择
<	检查邮件的总长度是否小于设置值
>	检查邮件的总长度是否大于设置值

在 action 部分，每一个行为的开头可以使用相应符号来执行不同的操作，常用的符号如表 10-7 所示。

表 10-7　action 的常用符号及其说明

符　号	说　明
!	将邮件转发到指定的地址
|	启动相应的程序
{}	括号之间可以再嵌套规则

除了以上的规则外，Procmailrc 配置文件中还包含很多环境变量信息，常用的环境变量如表 10-8 所示。

表 10-8　Procmailrc 的常用环境变量及其说明

环境变量	说　明
PATH	检索执行文件的路径
SENDMAIL	系统中 sendmail 的路径，也可以是 postifx 链接的 sendmail 路径
VERBOSE	打开或者关闭详细日志信息
LOGFILE	指定日志文件，默认为/var/log/procmail.log
ORGMAIL	用户的主目录。默认为/var/mail/$LOGNAME
DEFAULT	系统存放信箱的文件位置，默认和 ORGMAIL 相同
MAILDIR	Procmail 工作和执行的目录，默认为$HOME 目录

例如，如果想过滤掉来自 test@163.com 的所有邮件，那么通过配置 Procmail 可以使得来自 test@163.com 的邮件直接被送到 Linux 系统中的/dev/null 目录里。具体的配置可以参考如下：

```
:0
* ^From.* test@163.com
{
  :0
  /dev/null
}
```

经过上面的例子，读者应该能够了解 Procmail 规则的基本使用方法了。下面我们就给出一个综合的实例来分析 procmailrc 文件的组成和编写方法。

```
PATH=/bin:/sbin:/usr/bin:/usr/sbin
SHELL=/bin/bash
MAILDIR=$HOME/Mail
DEFAULT=$MAILDIR/inbox
LOGFILE="/var/log/procmail.log"

:0
 * ^From.*noreply@googlegroups.com
 comp.lang.c
 #将来自 noreply@googlegroups.com 的邮件保存到~/Mail/comp.lang.c （这是一个文本文件）
 :0
 * .*zeuux-universe@zeuux.org #这里没有用 From，那么所有抄送的邮件也会被过滤，就不会有遗漏了
 zeuux
 #将来自 zeuux-universe@zeuux.org 的邮件保存到~/Mail/zeuux(这是一个文本文件)
 :0
 * .*linux-kernel@zh-kernel.org #这里的“.*”代表任意多少字符串
 zh-kernel
 :0
 * ^From.*billgates@microsoft.com
#Billgates 的邮件放入/dev/null，实际上就是删除
 /dev/null
 :0 #最后的这个配置就是指如果上面分类剩下的信件全放入 inbox 里(inbox 是个文件夹，有三个子目录 new,tmp,cur)
 * .* inbox
 # 黑名单(垃圾邮件)
 :0:
 * ^From.*badguy
 /dev/null
```

关于 procmailrc 格式的具体参考，读者可以查看/usr/share/doc/procmail-3.22/examples/目录，其中包含多个规则的实例，读者可以根据这些实例和自己的情况进行修改，然后复制到相应的目录即可。另外，procmailrc 的详细使用手册还可以使用如下的命令来查看：

```
[root@CentOS /]# man procmailrc
[root@CentOS /]# man procmailex
```

10.6.4　Procmail 的启用

设置好邮件的过滤规则后，还需要让 Postfix 调用 Procmail 以进行邮件过滤。需要在 Posrfix 的配置文件 main.cf 中添加 mailbox_command 配置选项。此选项指明了 Postfix 的

本地进程将调用哪个命令分发本地邮件，默认为空，将其指定为 Procmail 后，则本地邮件会分发给 Procmail 程序来执行，那么之前配置的邮件过滤规则就生效了。具体的配置选项如下：

```
mailbox_command = /usr/bin/procmail
```

10.7 本 章 小 结

本章首先介绍了有关邮件系统的基本工作原理，然后介绍了几种重要的邮件协议，接着讲述了 Postfix 邮件服务器的安装方法及配置方法。最后，我们还介绍了 Postfix 与其他一些软件的配合使用方法。

10.8 课 后 习 题

1. 填空题

(1) Linux 下主流电子邮件服务器有：__________、__________、__________。
(2) Postfix 的主配置文件名为__________，其默认所在位置为__________。

2. 选择题

(1) 下列(　　)指令可以用来观看使用者邮件。
A. mail　　B. lynx　　C. ftp　　D. w
(2) 电子邮件使用的协议包括(　　)。
A. POP3　　B. IMAP　　C. SMTP　　D. RMVB

3. 简答题

(1) 什么是 MUA？什么是 MTA？
(2) 简述 Postfix 邮件服务器与其他邮件服务器相比有什么优势。

4. 操作题

(1) 按如下要求配置 Postfix:
① 配置/etc/postfix/main.cf，设定主机名称为 t1.ckhitler.org。
② 发信时显示出”发信源主机”项目，即邮件上面的 mail from 的那个位址。
③ 监控 postfix 的所有网卡。
④ 设定能收信的主机名称为 t1.ckhitler.org、localhost。
⑤ 规定信任的用户端为 127.0.0.0/8、192.168.6.0/24。
⑥ 规范 relay 的下一部 MTA 主机位址，指定为$mydestination。
⑦ 设定邮件别名为 alias_maps、alias_database，指定别名文件为/etc/postfix/aliases。
⑧ 重启服务。
⑨ 测试邮件收发。
(2) 安装 Postfix 服务器，并使用默认配置启动 Postfix 服务器。
(3) 为 Postfix 服务器添加账户，要求：用户名任意，此账户必须可以用来发送和接收邮件。

第 11 章

Linux 服务器安全技术

随着 Internet 规模的迅速扩大，安全问题也越来越重要，而构建防火墙是保护系统免受侵害的最基本的一种手段。虽然防火墙并不能保证系统绝对的安全，但由于它简单易行、工作可靠、适应性强，还是得到了广泛的应用。本章主要介绍与 Linux 系统紧密集成的 Iptables 防火墙的工作原理、命令格式，以及一些应用实例。

11.1 防火墙概述

防火墙是建立在内外网络边界上的过滤封锁机制。一般来说，内部网络被认为是安全和可信赖的，而外部网络被认为是不安全和不可信赖的。防火墙的作用是防止不希望的、未经授权的通信进出被保护的网络，迫使一个组织强化自己的网络安全策略，被认为是在可信的内部网络和不安全可信的外部网络之间提供的一个强化内部网络安全的政策。防火墙是不同网络之间信息的唯一出入口，能根据我们制定的策略来控制进出数据流，其本身具有抗攻击能力，能有效保证网络内部的安全。一个正确配置的防火墙可以极大地增加系统安全性。防火墙作为网络安全措施中一个重要的组成部分，一直受到人们的普遍关注。

Linux 操作系统为增加系统安全性而在内核级别提供了防火墙保护功能。Linux 的防火墙存在于你的计算机和网络之间，用来判定网络中的远程用户有权访问你的计算机上的资源。Linux 防火墙其实是操作系统本身所自带的一个功能模块。通过安装特定的防火墙内核，Linux 操作系统会对接收到的数据包按一定的策略进行处理。而用户所要做的，就是使用特定的配置软件(如 Iptables)去定制适合自己的“数据包处理策略”。

Linux 防火墙有如下特性。

(1) 防火墙包过滤：对数据包进行过滤可以说是任何防火墙所具备的最基本的功能，而 Linux 防火墙本身从某个角度也可以说是一种“包过滤防火墙”。在 Linux 防火墙中，操作系统内核对到来的每一个数据包进行检查，从它们的包头中提取出所需要的信息，如源 IP 地址、目的 IP 地址、源端口号、目的端口号等，再与已建立的防火规则逐条进行比较，并执行所匹配规则的策略，或执行默认策略。

值得注意的是，在制定防火墙过滤规则时通常有两个基本的策略方法可供选择：一个是默认允许一切，即在接受所有数据包的基础上明确地禁止那些特殊的、不希望收到的数据包；还有一个策略就是默认禁止一切，即首先禁止所有的数据包通过，然后再根据所希望提供的服务去一项项地允许需要的数据包通过。一般说来，前者使启动和运行防火墙变得更加容易，但却更容易为自己留下安全隐患。通过在防火墙外部接口处对进来的数据包进行过滤，可以有效地阻止绝大多数有意或无意的网络攻击，同时，对发出的数据包进行限制，可以明确地指定内部网中哪些主机可以访问互联网，哪些主机只能享用哪些服务或登录哪些站点，从而实现对内部主机的管理。可以说，在对一些小型内部局域网进行安全保护和网络管理时，包过滤确实是一种简单而有效的手段。

(2) 代理：Linux 防火墙的代理功能是通过安装相应的代理软件实现的。它使那些不具备公共 IP 的内部主机也能访问互联网，并且很好地屏蔽了内部网，从而有效保障了内部主机的安全。

(3) IP 伪装：IP 伪装(IP Masquerade)是 Linux 操作系统自带的又一个重要功能。通过在系统内核增添相应的伪装模块，内核可以自动地对经过的数据包进行“伪装”，即修改包头中的源目的 IP 信息，以使外部主机误认为该包是由防火墙主机发出来的。这样做，可以有效解决使用内部保留 IP 的主机不能访问互联网的问题，同时屏蔽了内部局域网。

11.2　Iptables 简介

Linux 在 2.0 的内核中,采用了 ipfwadm 来操作内核包过滤规则，它仅仅能分析 TCP、UDP、ICMP 协议，功能有限。在 Linux 的 2.2 的内核中,开始采用 ipchains 来控制内核包过滤规则。ipchains 重写了 ipfwadm 的包过滤系统，能够支持所有的基于 TCP/IP 的协议，但数据包处理过程繁琐效率较低。从 Linux 的 2.4 内核开始，采用一个全新的 Netfilter/Iptables 防火墙系统，取代了原有的 ipchains。

11.2.1　Netfilter/Iptables 工作原理

Netfilter/Iptables 是一个全新的 Linux 网络层防火墙，它完全改变了以往 Linux 防火墙的处理结构。

其中 Netfilter 是一个系统构架，它在内核中添加了 5 个位置的固定检查点(HOOK)，并为 IPv4 和 IPv6 在这些检查点上分别定义了一套钩子函数进行处理，实现如包过滤、NAT 或者是用户自定义的功能。当数据包通过这几个检查点的时候，这些钩子函数就会被调用，它们根据管理员添加的规则对数据包执行检测处理和过滤操作。IPv4 协议栈中的检查点位置如图 11-1 所示。

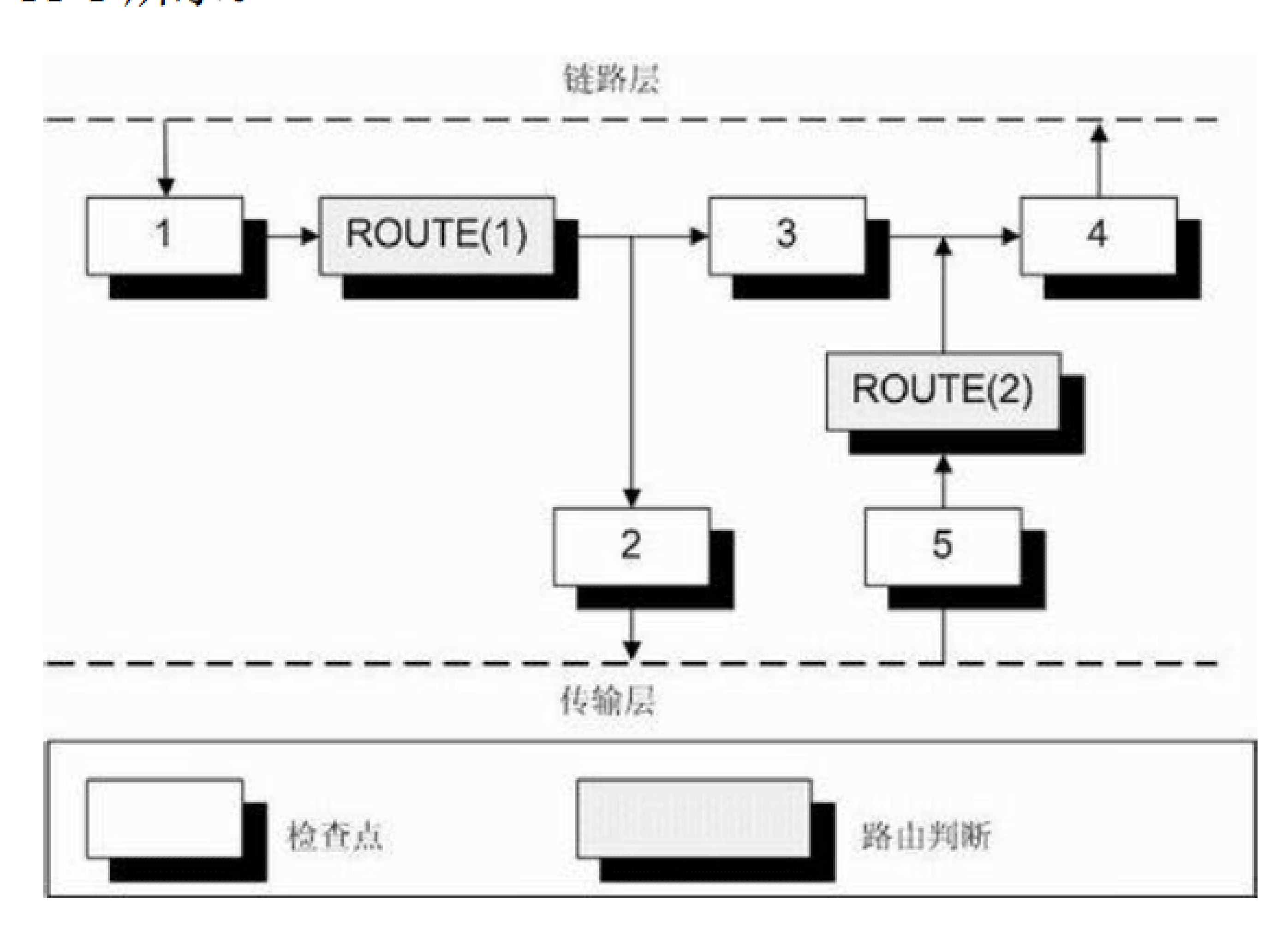


图 11-1　Netfilter 框架系统流程图

检查点所对应的钩子函数如图 11-2 所示。图中数据包从左上方进入系统，经过第一个钩子函数 NF_IP_PRE_ROUTING 的处理后进入路由模块。在路由模块中通过判断数据包的目的 IP 地址决定该数据包是需要转发还是发送给本机。若是发送给本机的，则数据包交给钩子函数 NF_IP_LOCAL_IN 处理，然后再传递给上层协议；若数据包应转发，则它会被 NF_IP_FORWARD 处理，再经过洗衣歌钩子函数 NF_IP_POST_ROUTING 处理后，交给下层协议封装，最后传输到网络上。而防火墙本身产生的数据经过 NF_IP_POST_ROUTING 处理后，交由下层协议封装，再次传输到网络上。

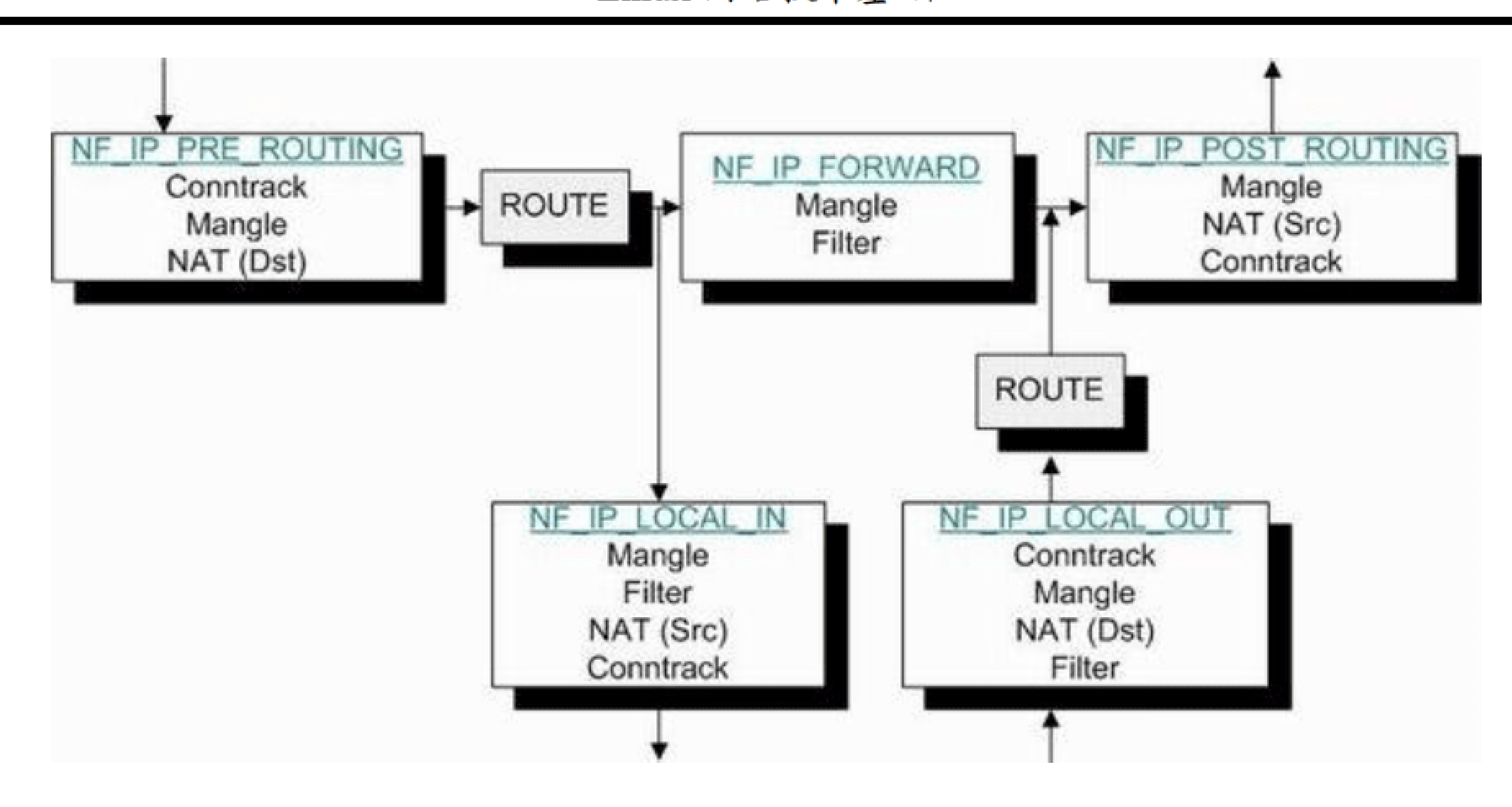


图 11-2 IPv4 检查点对应的钩子函数

在上述对数据包处理的过程中，这些钩子函数会调用 Iptables 工具加入的与这个函数相关的规则，对流经它的数据包进行处理，从而完成用户指定的网络安全防护任务。因此 Iptables 是一个管理工具，管理员通过它向 Netfilter 框架系统添加过滤规则，而运用这些规则对数据包进行处理的则是 Netfilter 框架系统。对于我们来说，重要的是掌握 Iptables 这个工具，学会利用它向内核添加规则。

11.2.2 Iptables 简介

Iptables 的作用就是添加 Netfilter 提供规则，这些规则告诉 Netfilter 对于从某些地方来、到某些地方去的或者具有某些特定特征的数据包采用什么样的动作。如果一个数据包与一条规则匹配，那么 Netfilter 上的钩子函数就会使用规则所指定的目标(ACCEPT、REJECT、DROP)允许该数据包通过或者阻塞该数据包。

在 Iptables 中，一条规则只定义某一种类型的数据包，要定义多种类型的数据包就需要使用多条规则。如果定义的多条规则均被同一钩子函数调用的话，那么就把被同一个钩子函数处理的多条规则称为一条规则链。Iptables 维护的 5 条规则链与 Netfilter 中定义的 5 个钩子函数一一对应，分别是：PREROUTING、FORWARD、POSTROUNTING、INPUT、OUTPUT。

也就是说，当使用 FORWARD 规则链设置的规则时，不管指定了多少条规则，它们都会被钩子函数 NF_IP_FORWARD 调用。同样对于其他的规则链也是一样。

上面提到，管理员利用 Iptables 添加规则，规则就是和数据包匹配的条件，每条规则后面还指定了目标，对符合规则的数据包采取什么样的动作。那么除了给规则设定目标外，还需要给指定的规则链设定默认的策略。数据包如何和规则链所有的规则都不匹配的话，应该对数据包采取什么样的动作，是丢弃还是放行。禁止一切的默认策略是推荐的方法。数据包和规则链中的规则匹配时，是自顶向下开始的，第一条规则如果和数据包不匹配，就和第二条规则匹配。如果与第二条数据包匹配，就执行第二条规则中定义的目标，而不用再和第三条规则去匹配了。图 11-3 给出了数据包匹配规则的流程图。

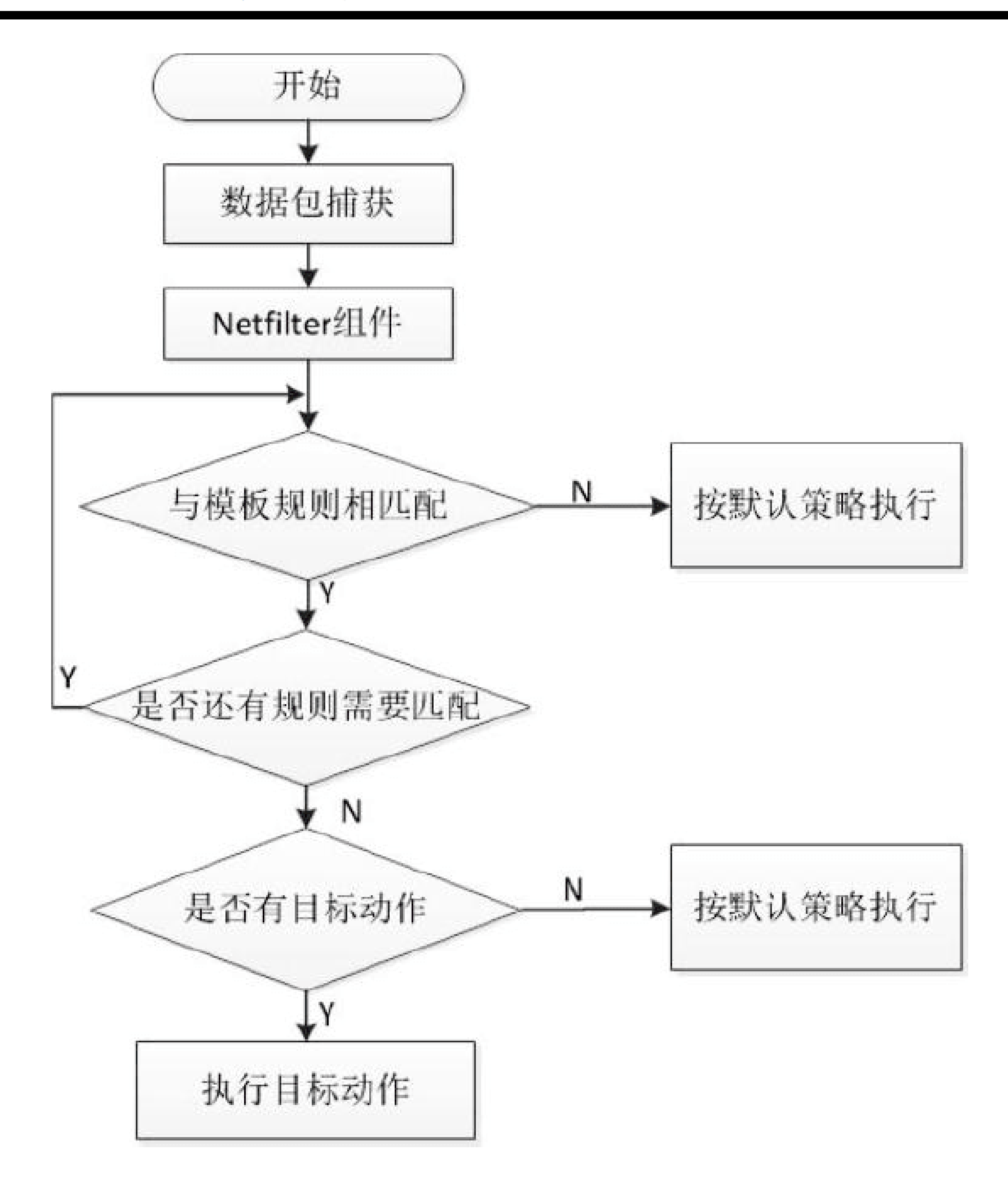


图 11-3　Iptables 规则匹配流程图

为方便管理，Iptables 规则被组织在单个不同的规则表中：filter、nat 和 mangle。其中 filter 表中的规则链主要用于数据包过滤，nat 表中规则链用于地址转换，mangle 表中规则链用于策略路由和特殊应用。以下是每张表中所包含的规则链：

- filter：INPUT、FORWARD、OUTPUT。
- nat：PREROUTING、POSTROUTING、OUTPUT。
- mangle：OUTPUT、POSTROUTING。

设定规则时，首先是指明你选用的表，然后制定表的规则链。没有指定表的情况下，系统默认使用 filter 表。

11.3　Iptables 的安装和配置

11.3.1　Iptables 的安装

在 CentOS 5 中，用户如果在系统安装时完全安装或者定制了防火墙套件，就会安装 Iptables 软件包。可以使用 rpm 查询命令查询是否已经安装了 Iptables 软件包，如下所示：

```
# rpm  -qa iptables
iptables-1.3.5-5.3.el5_4.1
```

如上述命令结果没有输入，则说明系统中没有安装 Iptables 软件包，则可以加载 CentOS 光盘后，进入到 CentOS 目录，用下面命令安装即可：

```
cd /media/CentOS*/CentOS*/
#rpm -ivh iptables*
```

当然，用户也可以选择通过 yum 安装或者通过下载源码进行编译安装。

11.3.2 Iptables 的启动和关闭

(1) 启动防火墙可以使用如下命令：

```
# service iptables start
```

或者

```
# /etc/rc.d/init.d/iptables start
```

显示结果如下所示：

```
应用 iptables 防火墙规则：                                    [确定]
载入额外 iptables 模块：ip_conntrack_netbios_ns                  [确定]
```

(2) 停止防火墙：

```
# service iptables stop
```

或者

```
# /etc/rc.d/init.d/iptables stop
```

显示结果如下所示：

```
清除防火墙规则：                                              [确定]
把 chains 设置为 ACCEPT 策略：filter                             [确定]
正在卸载 Iiptables 模块：                                      [确定]
```

(3) 重启防火墙：

```
# service iptables restart
```

或者

```
# /etc/rc.d/init.d/iptables start
```

(4) 设置防火墙开机自启动：

```
# chkconfig iptables on
```

关闭防火墙：

```
# service iptables stop
```

关闭防火墙开机自启动：

```
# chkconfig iptables off
```

11.3.3 Iptables 的配置文件

Iptables 的主要配置文件有两个，/etc/sysconfig/iptables 是用来保存 iptables 规则的文件，/etc/sysconfig/iptables-config 是用来存放 iptables 加载时所用的模块。

1. 查看/etc/sysconfig/iptables 配置文件内容

使用下面命令查看/etc/sysconfig/iptables 中内容：

```
# more /etc/sysconfig/iptables
```

显示 iptables 规则的结果如下所示：

```
# more /etc/sysconfig/iptables
# more /etc/sysconfig/iptables;
# Firewall configuration written by system-config-securitylevel
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

2. 保存当前 Iptables 规则

使用 Iptables 在命令行添加的规则能立即生效，但这些规则保存在内存中，如果重启 Iptables 或计算机，那么刚刚添加的规则都会丢失。如希望添加的规则在重启 Iptables 或计算机后依然生效，就应将这些规则保存在/etc/sysconfig/iptables 文件中。可以用以下命令实现：

```
#  service iptables save
```

或者运行

```
# iptables-save
```

输出显示结果如下：

```
#  service iptables save
将当前规则保存到 /etc/sysconfig/iptables:                [确定]
```

执行上面命令后，Iptables 会运行/sbin/iptables-save 程序，将规则写入到/etc/sysconfig/iptables 文件中。该命令只有管理员用户才能执行，因为 Iptables 文件的权限如下所示：

```
# ll iptables
-rw------- 1 root root 1538 03-22 09:27 iptables
```

3. 保存 Iptables 规则到文件

```
# iptables-save > iptables.bak
```

该命令用于将目前的 iptables 规则备份至 iptables.bak 文件中。

4. 从文件恢复 Iptables 规则

```
iptables-restore < iptables.bak
```

该命令用于将 iptables.bak 文件中的规则恢复到当前的 iptables 中。

11.4 Iptables 规则配置

从内核来看，规则就是决定如何处理一个语句。如果一个包符合所有的条件我们就运行目标或者调用自定义规则链。规则语法格式如下(如图 11-4 所示)：

```
iptables [-t 表名] <命令> [链名] [规则号] [规则] [-j 目标]
```

-t 选项用于指定所使用的表，Iptables 防火墙默认有 filter、nat 和 mangle 三张表，也可以是用户自定义的表。表中包含了分布在各个位置的链，iptables 命令所管理的规则就是存在于各种链中的。该选项不是必需的，如果未指定一个具体的表，则默认使用的是 filter 表。

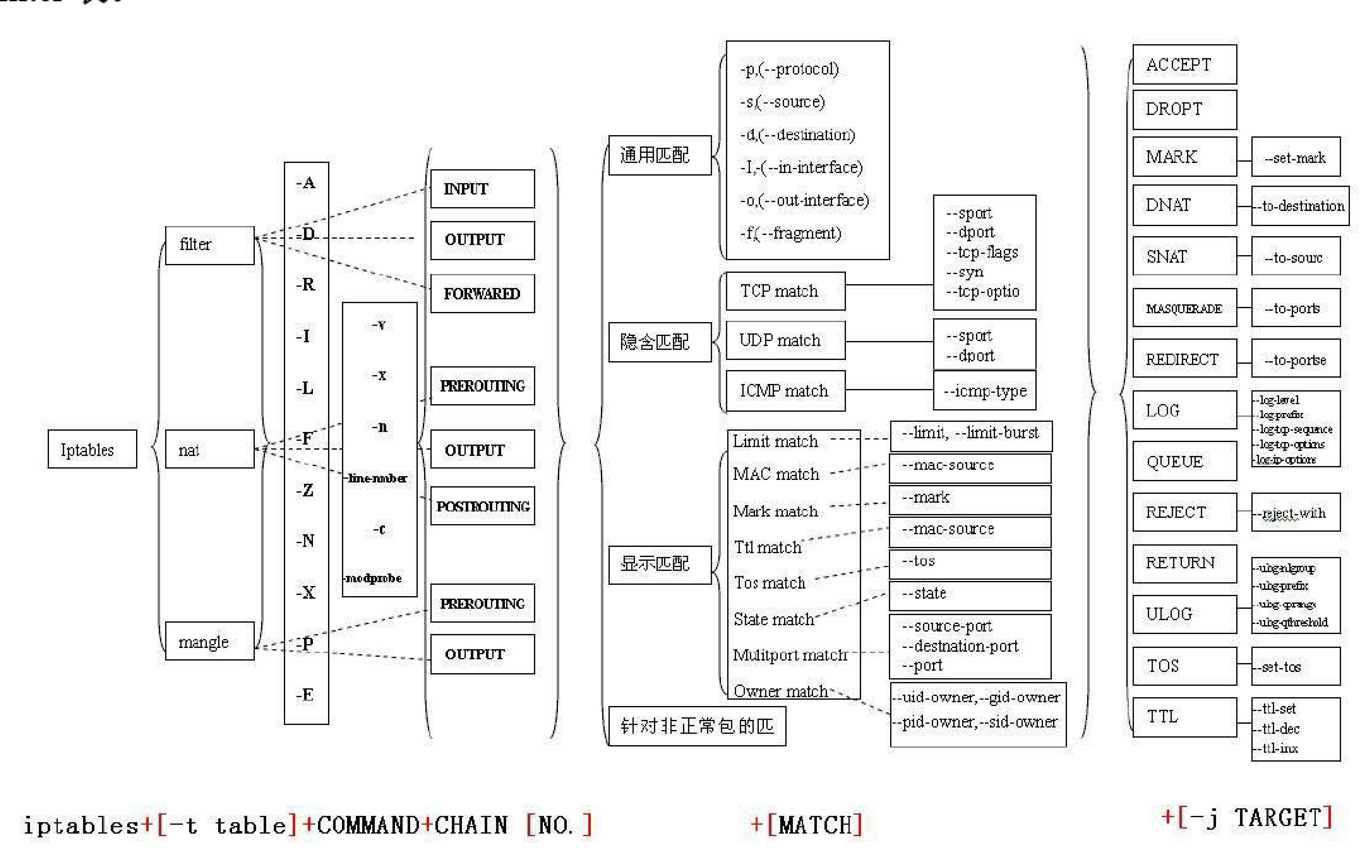


图 11-4 Iptables 的用法

Iptables 的命令选项是必须要有的，它告诉 Iptables 要做什么事情，是添加规则、修改规则还是删除规则。有些命令选项后面要指定具体的链名称，而有些可以省略，此时，是对所有的链进行操作。还有一些命令要指定规则号。具体的命令选项名称及其与后续选项的搭配形式如下所示。

示例 1：

```
-A <链名> <规则>
```

功能：在指定链的末尾添加一条或多条规则。

示例 2：

```
-D <链名> <规则> -D <链名> <规则号>
```

功能：从指定的链中删除一条或多条规则。可以按照规则的序号进行删除，也可以删除满足匹配条件的规则。

示例 3：

```
-R <链名> <规则号> <规则>
```

功能：在指定的链中用新的规则置换掉某一规则号的旧规则。

示例 4：

```
-I <链名> [规则号] <规则>
```

功能：在给出的规则序号前插入一条或多条规则，如果没有指定规则号，则默认是 1。

示例 5：

```
-L [链名]
```

功能：列出指定链中的所有规则，如果没有指定链，则所有链中的规则都将被列出。

示例 6：

```
-F [链名]
```

功能：删除指定链中的所有规则，如果没有指定链，则所有链中的规则都将被删除。

示例 7：

```
-N <链名>
```

功能：建立一个新的用户自定义链。

示例 8：

```
-X [链名]
```

功能：删除指定的用户自定义链，这个链必须没有被引用，而且里面也不包含任何规则。如果没有给出链名，这条命令将试着删除每个非内建的链。

示例 9：

```
-P <链名> <目标>
```

功能：为指定的链设置规则的默认目标，当一个数据包与所有的规则都不匹配时，将采用这个默认的目标动作。

示例 10：

```
-E <旧链名> <新链名>
```

功能：重新命名链名，对链的功能没有影响。

以上是有关 Iptables 命令格式中有关命令选项部分的解释。Iptables 命令格式中的规则部分由很多选项构成，主要指定一些 IP 数据包的特征。例如，上一层的协议名称、源 IP 地址、目的 IP 地址、进出的网络接口名称等，下面列出构成规则的常见选项。

- -p<协议类型>：指定上一层协议，可以是 icmp、tcp、udp 和 all。
- -s<IP 地址/掩码>：指定源 IP 地址或子网。
- -d<IP 地址/掩码>：指定目的 IP 地址或子网。
- -i<网络接口>：指定数据包进入的网络接口名称。
- -o<网络接口>：指定数据包出去的网络接口名称。

上述选项可以进行组合，每一种选项后面的参数前可以加“!”，表示取反。

对于-p 选项来说，确定了协议名称后，还可以有进一步的子选项，以指定更细的数据包特征。常见的子选项如下所示。

- -p tcp --sport <port>：指定 TCP 数据包的源端口。
- -p tcp --dport <port>：指定 TCP 数据包的目的端口。
- -p tcp --syn：具有 SYN 标志的 TCP 数据包，该数据包要发起一个新的 TCP 连接。
- -p udp --sport <port>：指定 UDP 数据包的源端口。
- -p udp --dport <port>：指定 UDP 数据包的目的端口。
- -p icmp --icmp-type <type>：指定 icmp 数据包的类型，可以是 echo-reply、echo-request 等。

上述选项中，port 可以是单个端口号，也可以是以 port1:port2 表示的端口范围。每一选项后的参数可以加“!”，表示取反。

上面介绍的这些规则选项都是 Iptables 内置的，Iptables 软件包还提供了一套扩展的规则选项。使用时需要通过-m 选项指定模块的名称，再使用该模块提供的选项。下面列出几个模块名称和其中的选项，大部分的选项也可以通过“!”取反。

- -m multiport --sports <port, port, …> 功能：指定数据包的多个源端口，也可以以 port1:port2 的形式指定一个端口范围。
- -m multiport --dports <port, port, …> 功能：指定数据包的多个目的端口，也可以以 port1:port2 的形式指定一个端口范围。
- -m multiport --ports <port, port, …> 功能：指定数据包的多个端口，包括源端口和目的端口，也可以以 port1:port2 的形式指定一个端口范围。
- -m state --state <state> 功能：指定满足某一种状态的数据包，state 可以是 INVALID、ESTABLISHED、NEW 和 RELATED 等，也可以是它们的组合，用“,”分隔。
- -m connlimit --connlimit-above <n> 功能：用于限制客户端到一台主机的 TCP 并发连接总数，n 是一个数值。
- -m mac --mac-source <address> 功能：指定数据包的源 MAC 地址，address 是 xx:xx:xx:xx:xx:xx 形式的 48 位数。
- -m 选项可以提供的模块名和子选项内容非常多，为 Iptables 提供了非常强大、细致的功能，所有的模块名和子选项可以通过“man iptables”命令查看 Iptables 命令的手册页获得。

最后，Iptables 命令中的-j 选项可以对满足规则的数据包执行指定的操作，其后的“目标”可以是以下内容。

- -j ACCEPT：将与规则匹配的数据包放行，并且该数据包将不再与其他规则匹配，而是跳向下一条链继续处理。
- -j REJECT：拒绝所匹配的数据包，并向该数据包的发送者回复一个 ICMP 错误通知。该处理动作完成后，数据包将不再与其他规则匹配，而且也不跳向下一条链。
- -j DROP：丢弃所匹配的数据包，不回复错误通知。该处理动作完成后，数据包将不再与其他规则匹配，而且也不跳向下一条链。
- -j REDIRECT：将匹配的数据包重定向到另一个位置，该动作完成后，会继续与其他规则进行匹配。
- -j LOG：将与规则匹配的数据包的相关信息记录在日志(/var/log/message)中，并继续与其他规则匹配。
- -j <规则链名称>：数据包将会传递到另一规则链，并与该链中的规则进行匹配。

除了上述目标动作外，还有一些与 NAT 有关的目标，将在后面章节中讲述。所有的目标也可以通过查看 Iptables 命令的手册页获得。

11.5 防火墙规则设定

11.5.1 Linux 防火墙的默认规则

在 Linux 系统中，可以通过使用 Iptables 命令构建各种类型的防火墙。CentOS 5 操作系统默认安装时，Iptables 防火墙已经安装，并且开机后会自动添加了一些规则，这些规则实际上是由/etc/sysconfig 目录中的 Iptables 文件决定的。可以通过“iptables –L”命令查看这些规则，如果用户未在防火墙打开端口，结果将显示如下所示。

```
# iptables -L --line-number
Chain INPUT (policy ACCEPT)                   //INPUT 规则链
num  target     prot opt source               destination
1    RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)                 // FORWARD 规则链
num  target     prot opt source               destination
1    RH-Firewall-1-INPUT  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)                  // OUTPUT 规则链
num  target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)      // 自定义规则链
num  target     prot opt source               destination
1    ACCEPT     all  --  anywhere             anywhere
2    ACCEPT     icmp --  anywhere             anywhere            icmp any
3    ACCEPT     esp  --  anywhere             anywhere
4    ACCEPT     ah   --  anywhere             anywhere
5    ACCEPT     udp  --  anywhere             224.0.0.251         udp dpt:mdns
6    ACCEPT     udp  --  anywhere             anywhere            udp dpt:ipp
7    ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:ipp
8    ACCEPT     all  --  anywhere             anywhere            state
RELATED,ESTABLISHED
9    REJECT     all  --  anywhere             anywhere            reject-with
icmp-host-prohibited
```

由上面结果可知，iptables -L 默认列出的是 filter 表中的规则链，filter 表中总共有 4 条规则链。其中，INPUT、FORWARD 和 OUTPUT 链是内置的，而 RH-Firewall-1-INPUT 链是可由用户自己添加的自定义规则链。

1. 规则列的含义

由上面返回结果看出，在防火墙规则中每一条规则列出了 5 项内容：

- target 列表示规则的动作目标。
- prot 列表示该规则指定的上层协议名称，all 表示所有的协议。
- opt 列出了规则的一些选项。
- source 列表示数据包的源 IP 地址或子网，
- destination 列表示数据包的目的 IP 地址或子网，anywhere 表示所有的地址。

除了上述 5 项以外，如果存在，每一条规则的最后还要列出一些子选项。

如果执行 iptables 命令时加了-v 选项，则还可以列出每一条规则当前匹配的数据包数、字节数，以及要求数据包进来和出去的网络接口。如果加上-n 选项，则不对显示结果中的 IP 地址和端口做名称解析，直接以数字的形式显示。

2. 规则的解释

INPUT 链中的第 1 条规则 target 列的内容是 RH-Firewall-1-INPUT，opt 列是 all，source 和 destination 列均为 anywhere，表示所有的数据包都交给自定义的 RH-Firewall-1-INPUT 链去处理。FORWARD 链的第 1 条规则 2 与规则 1 完全一样。OUTPUT 链中没有规则。

RH-Firewall-1-INPUT 规则链中，列出了很多的规则，第 1 条规则表示接收所有的数据包。需要注意的是，如果在 Iptables 中加-v 选项列出这条规则时，将会看到 in 列是 lo，即要求数据包是从环回接口中进来的，而不是任意网络接口进来的数据包都接收。第 2 条规则表示接收所有 icmp 数据包，而且在 OUTPUT 链中没有规则，即发送数据包不会受限，因此本机的 ICMP 回复数据包也能正常回复，即 ping 命令可以正常使用。第 3、4 条规则表示接收所有的 esp 和 ah 协议的数据包，这两种协议属于 IPv6 协议。第 5 条规则表示目的地址是 224.0.0.251，目的端口是 mdns 的 UDP 数据包允许通过，224.0.0.251 是一种组播地址，mdns 是端口号的一种名称，如果执行 Iptables 命令时加了-n 选项，则会显示数字 5353，它是组播地址的 DNS 端口。第 6、7 条规则表示允许所有目的端口是 ipp 的 UDP 和 TCP 数据包通过，ipp 是端口 631 的名称解析，它是用于网络打印服务的端口。第 8 条规则表示所有状态是 RELATED 和 ESTABLISHED 的数据包通过，RELATED 状态表示数据包要新建一个连接，而且这个要新建的连接与现存的连接是相关的，如 FTP 的数据连接。ESTABLISHED 表示本机与对方建立连接时，对方回应的数据包，表明双方已经建立了数据链路。第 9 条规则表示拒绝所有的数据包，并向对方回应 icmp-host-prohibited 数据包。

3. 规则的顺序

需要再次提醒的是，这些规则是有次序的。当一个数据包进入 RH-Firewall-1-INPUT 链后，将依次与规则 1 至规则 8 进行比较。按照这些规则的目标设置，如果数据包能与规

则 1 至 9 中的任一条匹配，则该数据包将被接收。如果都不能匹配，则肯定能和规则 9 匹配，于是数据包被拒绝。

11.5.2　Linux 防火墙规则操作方法

防火墙功能可以应用在多种位置，可以安装在某一台主机上，主要用于保护主机本身的安全；可以安装在网络中的某一节点，专门用于保护网络中其他计算机的安全；也可以为内网的客户机提供 NAT 服务，使内网的客户机共用一个公网 IP，以便节省 IP 地址资源。下面首先介绍一下主机防火墙的应用示例。

当一台服务器为外界提供比较重要的服务，或者一台客户机在不安全的网络环境中使用时，都需要在计算机上安装防火墙，以最大限度地防止主机受到外界的攻击。用户可以根据自己主机的功能关闭已经开放的端口，或者开放更多的端口，以便允许符合更多规则的数据包通过。

1. 在用户自定义规则链中添加规则

例如，为了使主机能为外界提供 telnet 服务，除了配置好 telnet 服务器外，还需要开放 TCP 的 23 号端口。因为在默认的防火墙配置中，并不允许目的端口为 23 的 TCP 数据包进入主机。为了开放 TCP23 号端口，可以有两种办法，一种是在 RH-Firewall-1-INPUT 链中加入相应的规则，还有一种是把规则加到 INPUT 链中。但需要注意的是，规则是有次序的，如果使用以下命令，则是没有效果的。

```
# iptables -A RH-Firewall-1-INPUT -p tcp --dport 23 -j ACCEPT
```

上述命令执行后，可以再次查看规则情况。

```
# iptables -L -n --line-number
…
Chain RH-Firewall-1-INPUT (2 references)
num  target      prot opt  source           destination
… //省略
11   ACCEPT     tcp  --  0.0.0.0/0        0.0.0.0/0    state NEW tcp dpt:80
12   ACCEPT     tcp  --  0.0.0.0/0        0.0.0.0/0    state NEW tcp dpt:25
13   REJECT     all  --  0.0.0.0/0        0.0.0.0/0    reject-with icmp-host-
prohibited
14   ACCEPT     tcp  --  0.0.0.0/0        0.0.0.0/0 tcp dpt:23
```

可以看到，新添加的规则位于最后的位置。由于所有的数据包都可以与目标动作为 REJECT 的规则号为 13 的规则匹配，而 REJECT 代表的是拒绝，因此数据包到达新添加的规则前肯定已被丢弃，这条规则是不会被使用的。为了解决这个问题，需要把上述规则插入到现有的规则中，要位于规则 13 的前面。下面是正确地开放 TCP23 号端口的命令。

```
# iptables -I RH-Firewall-1-INPUT 11 -p tcp --dport 23 -j ACCEPT
```

以上命令中，“-I RH-Firewall-1-INPUT 11”表示在 RH-Firewall-1-INPUT 链原来的规则 11 前面插入一条新规则，规则内容是接收目的端口为 23 的 TCP 数据包。

如果希望新加的规则与原来的规则 11、12 等类似，可以执行以下命令。

```
# iptables -I RH-Firewall-1-INPUT 11 -m state --state NEW -p tcp --dport
23 -j ACCEPT
```

以上是在 RH-Firewall-1-INPUT 链中添加规则，以开放 TCP23 号端口。

2. 在 INPUT 规则链添加规则

还有一种开放 TCP23 号端口的方法是在 INPUT 链中添加规则，具体命令如下所示。

```
# iptables -I INPUT 1 -p tcp --dport 23 -j ACCEPT
# iptables -L --line-number
Chain INPUT (policy ACCEPT)
num  target            prot opt    source          destination
1    ACCEPT                tcp   --    anywhere    anywhere    tcp dpt:telnet
2    RH-Firewall-1-INPUT    all   --    anywhere    anywhere
…
```

注意：添加的规则也要位于原来规则 2 的前面，否则，任何数据包都匹配规则 2，将会跳到 RH-Firewall-1-INPUT 链，并且不再回来。因此，添加在规则 2 后面的规则都是无效的。

3. 删除防火墙规则

如需要删除前面无效规则，可以执行以下命令。

```
# iptables -D RH-Firewall-1-INPUT 11
```

在上面的命令中，11 是 RH-Firewall-1-INPUT 规则表中规则的序号，可根据用户的具体要求加以改变。

4. 清空防火墙规则

在很多的时候，用户可能希望从最初的状态开始，构建自己的防火墙。为了从零开始设置 iptables 防火墙，可以用以下命令清空防火墙中所有的规则。

```
# iptables -F
```

然后再根据要求，添加自己的防火墙规则。一般情况下，保护防火墙所在主机的规则都添加在 INPUT 内置链中，以挡住外界访问本机的部分数据包。本机向外发送的数据包只经过 OUTPUT 链，一般不予限制。如果不希望本机为外界数据包提供路由转发功能，可以在 FORWARD 链中添加一条拒绝一切数据包通过的规则，或者干脆在内核中设置不转发任何数据包。

11.5.3 Linux 防火墙规则操作示例

当设置主机防火墙时，一般采取先放行，最后全部禁止的方法。也就是说，根据主机的特点，规划出允许进入主机的外界数据包，然后设计规则放行这些数据包。如果某一数据包与放行数据包的规则都不匹配，则与最后一条禁止访问的规则匹配被拒绝进入主机。下面列出一些主机防火墙中常用的 iptables 命令及其解释，这些命令添加的规则都放在 filter 表的 INPUT 链中。

示例 1：

```
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

功能：允许目的端口为 80 的 TCP 数据包通过 INPUT 链。

说明：这种数据包一般是用来访问主机的 Web 服务，如果主机以默认的端口提供 Web 服务，应该用这条规则开放 TCP80 端口。

示例 2：

```
iptables -A INPUT -s 192.168.1.0/24 -i eth0 -j DROP
```

功能：从接口 eth0 进来的、源 IP 地址的前 3 字节为 192.168.1 的数据包予以丢弃。

说明：需要注意这条规则的位置，如果匹配这条规则的数据包同时也匹配前面的规则，而且前面的规则是放行的，则这条规则对匹配的数据包将不起作用。

示例 3：

```
iptables -A INPUT -p udp --sport 53 --dport 1024:65535 -j ACCEPT
```

功能：在 INPUT 链中允许源端口号为 53，目标端口号为 1024 至 65535 的 UDP 数据包通过。

说明：这种特点的数据包是当本机查询 DNS 时，DNS 服务器回复的数据包。

示例 4：

```
iptables -A INPUT -p tcp --tcp-flags SYN,RST,ACK SYN -j ACCEPT
```

功能：SYN、RST、ACK 三个标志位中 SYN 位为 1，其余两个为 0 的 TCP 数据包予以放行。符合这种特征的数据包是发起 TCP 连接的数据包。

说明："--tcp-flags"子选项用于指定 TCP 数据包的标志位，可以有 SYN、ACK、FIN、RST、URG 和 PSH 共 6 种。当这些标志位作为"--tcp-flags"的参数时，用空格分成两部分。前一部分列出有要求的标志位，用","分隔；后一部分列出要求值为 1 的标志位，如果有多个，也用","分隔，未在后一部分列出的标志位其值要求为 0。

注意：这条命令因为经常使用，可以用"--syn"代替"--tcp-flags SYN,RST,ACK SYN"。

示例 5：

```
iptables -A INPUT -p tcp -m multiport --dport 20:23,53,80,110 -j ACCEPT
```

功能：接收目的端口为 20 至 23、53、80 和 110 号的 TCP 数据包。

说明："-m multiport"用于指定多个端口，最多可以有 15 项，用","分隔。

示例 6：

```
iptables -A INPUT -p icmp -m limit --limit 6/m --limit-burst 8 -j ACCEPT
```

功能：限制 ICMP 数据包的通过率，当一分钟内通过的数据包达到 8 个时，触发每分钟通过 6 个数据包的限制条件。

说明：以上命令中，除了 m 表示分以外，还可以用 s(秒)、h(小时)和 d(天)。这个规则主要用于防止 DoS 攻击。

示例 7：

```
iptables -A INPUT -p udp -m mac --mac-source ! 00:11:22:33:44:55 -j DROP
```

功能：拒绝源 MAC 地址不是 00:11:22:33:44:55 的 UDP 数据包。

说明：该规则不应该放在前面，否则，大部分的 UDP 数据包都将被拒绝，随后的规则将不会使用。

11.5.4 使用图形界面管理防火墙规则

在 CentOS 5 中，为了使用户能方便快捷构建 Iptables 主机防火墙，系统提供了配置主机防火墙的图形界面。在 CentOS 5 的 GNOME 桌面环境下，选择“系统”→“管理”→“安全级别和防火墙”命令后，将出现图 11-5 所示的对话框。

在安全级别管理设置中，有两个选项卡，分别是防火墙选项和 SELinux 选项。在对防火墙设置之前，用户应首先确认防火墙的状态。

在“防火墙选项”中，选择“防火墙”后面的下拉菜单，可以选择防火墙的状态是“启用”或者“禁用”，单击“应用按钮即可生效”。禁用防火墙后系统将无法得到有效防护，因此一般情况下，建议用户开机 Linux 自带的防火墙功能。

在“信任的服务”的列表框中已经列出了常见的网络服务名称，前面选中的服务所对应的网络端口是开放的，允许外界的用户访问。如果用户需要开放更多的服务，可以选中要开放的服务名称，再单击“应用”按钮即可。

如用户需要在防火墙的信任列表中将其他特定端口添加或者删除，可以单击窗口下方的“其他端口”文字，在防火墙中已经添加的其他端口及使用的网络传输协议将在下方列表框中显示出来。同时在右侧显示出“添加”和“删除”按钮，用于添加和删除“其他端口”列表框中列出的端口。

如果单击“添加”按钮，将出现图 11-6 所示的对话框。此时，可以输入需要开放的端口号，并选择 TCP 或 UDP 协议，然后单击“确定”按钮，在“其他端口”列表框中将出现所添加的端口号和协议名称。单击“删除”按钮可以删除列表框中选中的端口。为了使添加或删除端口生效，需要单击图 11-6 中的“应用”按钮，此时将出现图 11-7 所示的对话框，要求用户确认该操作，单击“是”按钮即可。

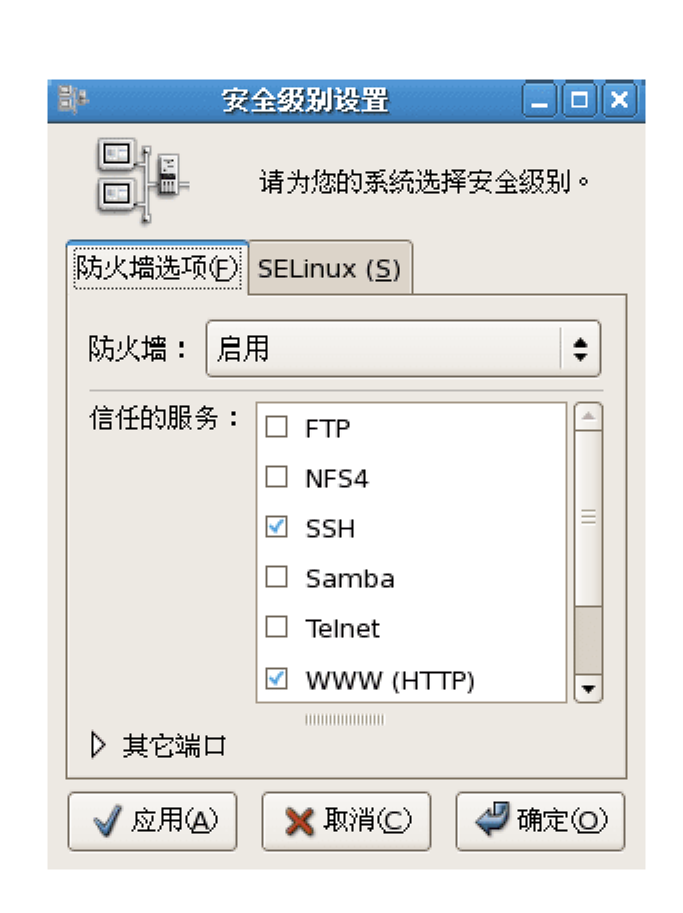


图 11-5 Linux 安全级别设置



图 11-6 为防火墙添加端口

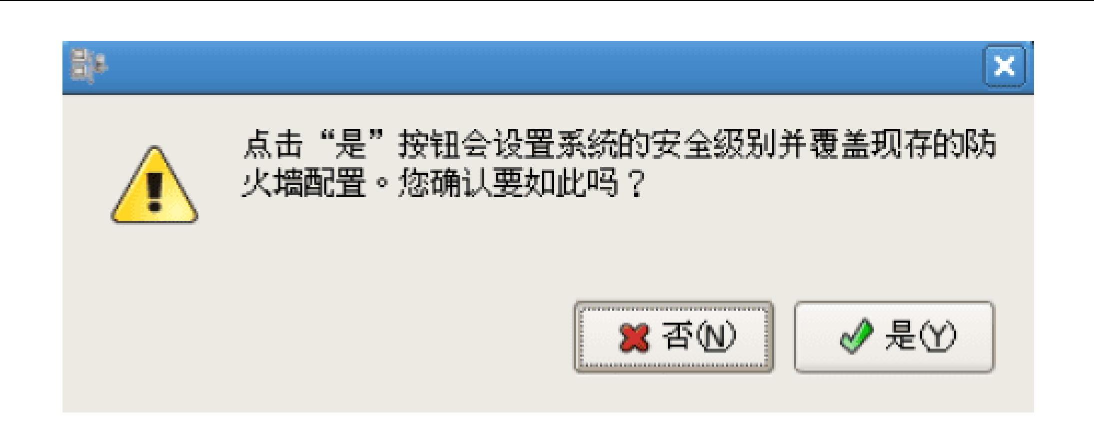


图 11-7　确认修改防火墙设置

以上是通过图形界面管理主机防火墙规则，实际的结果和命令方式是一样的。例如，如果刚才在图形窗口中将“信任的服务”设为 www(HTTP)，“其他端口”设为 8080 端口并选择 TCP 协议，然后再到终端查看防火墙中的规则时，将会发现如下结果。

```
# more /etc/sysconfig/iptables
# 显示结果如下所示:
# Firewall configuration written by system-config-securitylevel
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m tcp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j
ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -
j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

从以上结果可以看出，与初始的设置相比，多出了两条规则：

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j
ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j
```

也就是说，刚才的图形界面操作相当于输入了以下命令。

```
#iptables -I RH-Firewall-1-INPUT 13 -m state --state NEW -m tcp -p tcp -
-dport 80 -j ACCEPT
#iptables -I RH-Firewall-1-INPUT 13 -m state --state NEW -m tcp -p tcp -
-dport 8080 -j ACCEPT
```

通过刚刚的操作用户可以看出，CentOS 5 提供的防火墙图形界面管理功能非常有限，远不如采用命令行方式功能强大。

11.6 使用 Iptables 实现 NAT

11.6.1 NAT 概述

网络地址转换(NAT，Network Address Translation)广泛应用于网络中，它成功地解决了 IP 地址资源不足的问题，而且还能有效避免来自外部网络的攻击，隐藏并保护了网络内部计算机。它根据 RFC1631 开发的 IETF 标准，允许一个私有 IP 地址域以公有 IP 地址的方式使用 Internet。

借助 NAT，使用私有 IP 地址的内部网络通过路由器或防火墙向外发送数据时，私有 IP 地址会被转换为合法的公有 IP 地址，而当返回的数据包到达路由器或防火墙时，再将公有 IP 转换为私有 IP 地址。所以一个局域网只需要有少量合法 IP 即可实现多台机器联网的需求。NAT 极大程度上解决了 IPv4 地址匮乏的问题。

11.6.2 私有 IP 地址

IANA(The Internet Assigned Numbers Authority，互联网数字分配机构)将 IPv4 地址分为 A、B、C、D、E 5 类地址，其中 D 类地址用于组播，E 类地址用于科学研究，这两类地址不能在 Internet 上使用。故 Internet 上面能够分配给主机使用的 IP 地址只有 A、B、C 这三类地址。为了满足网络内部局域网用户使用 Internet 的需求，IANA 规定从 A、B、C 类地址中分别划出一部分地址供内部网络使用，这部分地址就是私有地址。私有地址默认在 Internet 上面是没有路由的，即在 Internet 上是无法通信的。私有地址的 IP 地址范围如表 11-1 所示。

表 11-1　私有 IP 地址范围

地址类型	私有地址网络名	子网掩码	IP 地址范围
A	10.0.0.0	255.0.0.0	10.0.0.1~10.255.255.254
B	172.16.0.0	255.240.0.0	172.16.0.1~172.31.255.254
C	192.168.0.0	255.255.0.0	192.168.0.1~192.168.255.254

11.6.3 NAT 的类型

NAT 实现有三种方式，分别是：静态地址转换、动态地址转换、网络地址端口转换。

- 静态地址转换就是在将内部网络的私有网络地址转换为合法的公有地址时，私有地址和公有地址是一对一的对应关系。静态地址转换一般应用于外部网络对内部网络中某些服务器的访问。
- 动态地址转换是指在外部网络上定义一系列的合法 IP 地址，在将内部网络的私有地址转换为公有地址时，私有地址和公有地址的对应关系是不确定的、随机的。每一个需要连接外部网络的内部主机都能分配到一个临时的公有地址，当用户断开连接时，这个公有地址就会被释放，留给其他主机使用。

- 网络地址端口转换也叫端口多路复用，是指将内部网络中所有的私有地址都转换为同一个合法的 IP 地址，然后再随机分配一个由 NAT 设备指定的不同的端口号。当多个内部 IP 地址映射到同一个合法地址的时候，使用端口号对其进行区分。采用这种方式可以最大程度地节约 IP 地址资源，是目前网络中使用最多的一种转换方式。

11.6.4　NAT 的工作原理

NAT 是定义于 RFC1631 中的 Internet 标准，主要用来使原本无法上网但可以使用内部 IP 地址的主机连接到 Internet 上。NAT 的出现大大减少了 IP 地址需求，可实现整个私有网络通过少量甚至一个公有 IP 地址连接 Internet，如图 11-8 所示。

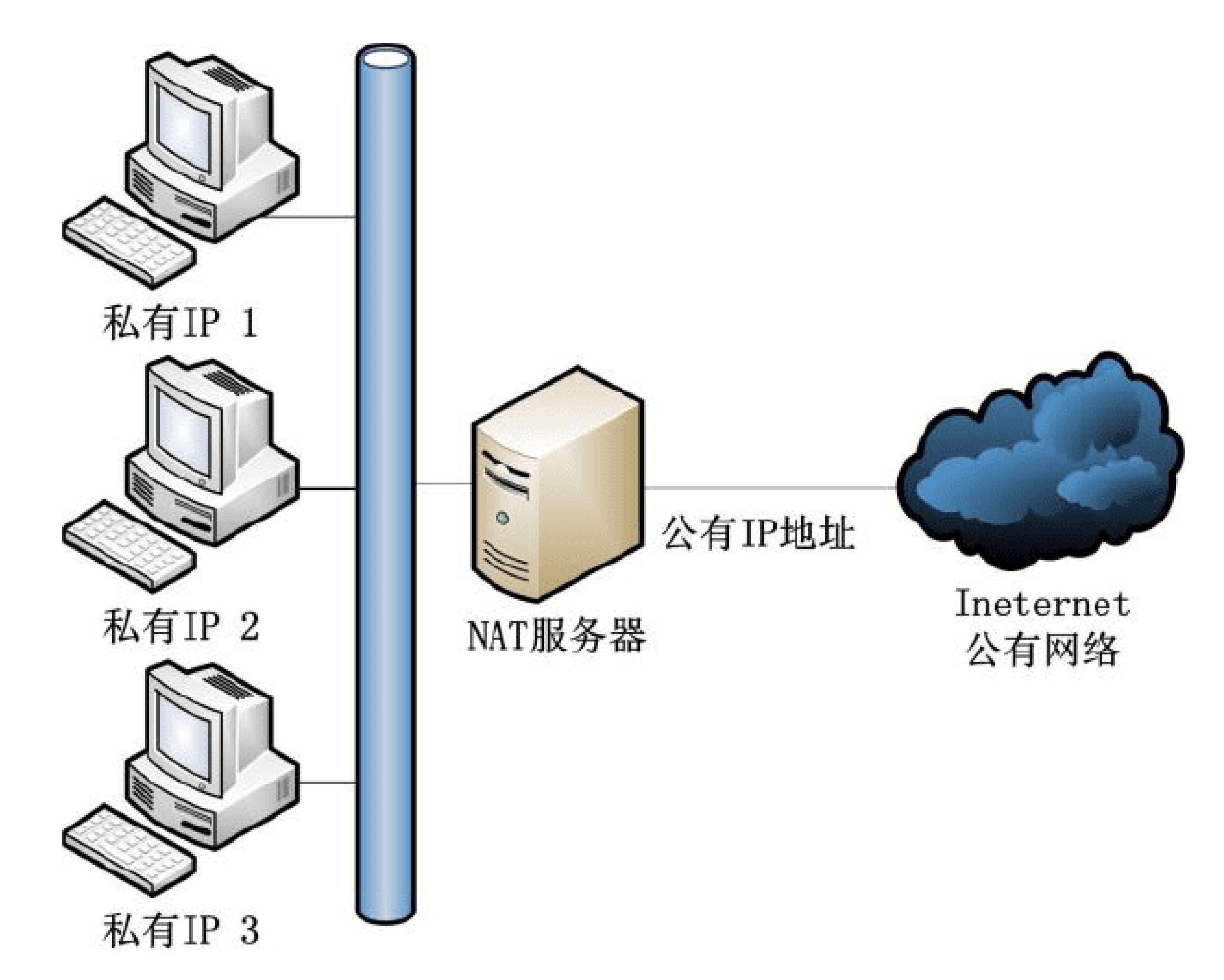


图 11-8　NAT 连接内部网络及 Internet

在 Netfilter/Iptables 中，将 NAT 分为两种类型，即源地址转换(SNAT)和目的地址转换(DNAT)。SNAT 是改变转发数据包的源地址，DNAT 是改变转发数据包的目的地址。

在前面提到过，为管理方便，Iptables 规则被组织在 filter、nat 和 mangle 三张不同的表里面，其中 NAT 表用来实现地址转换。NAT 表定义了 PREROUTING、POSTROUTING、OUTPUT 三条规则链，利用这三条规则链定义的规则会分别被 NF_IP_PRE_ROUTING、NF_IP_PSOT_ROUTING 和 NF_IP_LOCAL_OUT 这三个钩子函数调用。

利用 PREROUTING 规则链可以实现目的地址转。利用 POSTROUTING 规则链可以实现源地址转换和伪装。伪装是网络端口转换的一个特例，就是将所有的私有地址都映射到一个不固定的合法地址上。OUTPUT 规则链可以实现对防火墙本身产生的数据包进行地址转换。

11.6.5　源 NAT 配置案例

在前面的有关章节中已经介绍了路由和过滤数据包的方法，它们都不牵涉到对数据包

的 IP 地址进行改变。但源 NAT 需要对内网出去的数据包的源 IP 地址进行转换，用公网 IP 代替内网 IP，以便数据包能在 Internet 上传输。Iptables 的源 NAT 的配置应该是在路由和网络防火墙配置的基础上进行的。

Iptables 防火墙中有 3 张内置的表，其中的 nat 表实现了地址转换的功能。nat 表包含 PREROUTING、OUTPUT 和 POSTROUTING 3 条链，里面包含的规则指出了如何对数据包的地址进行转换。其中，源 NAT 的规则在 POSTROUTING 链中定义。这些规则的处理是在路由完成后进行的，可以使用“-j SNAT”目标动作对匹配的数据包进行源地址转换。

假设让 iptables 防火墙承担 NAT 服务器功能将内网 10.16.0.0/24 出去的数据包其源 IP 地址都转换外网接口 eth0 的公网 IP 地址 202.99.166.1(见图 11-9)，则需要执行以下 iptables 命令。

```
# iptables -t nat -A POSTROUTING -s 10.16.0.0/24 -o eth0 -j SNAT --to-source 202.99.166.1
```

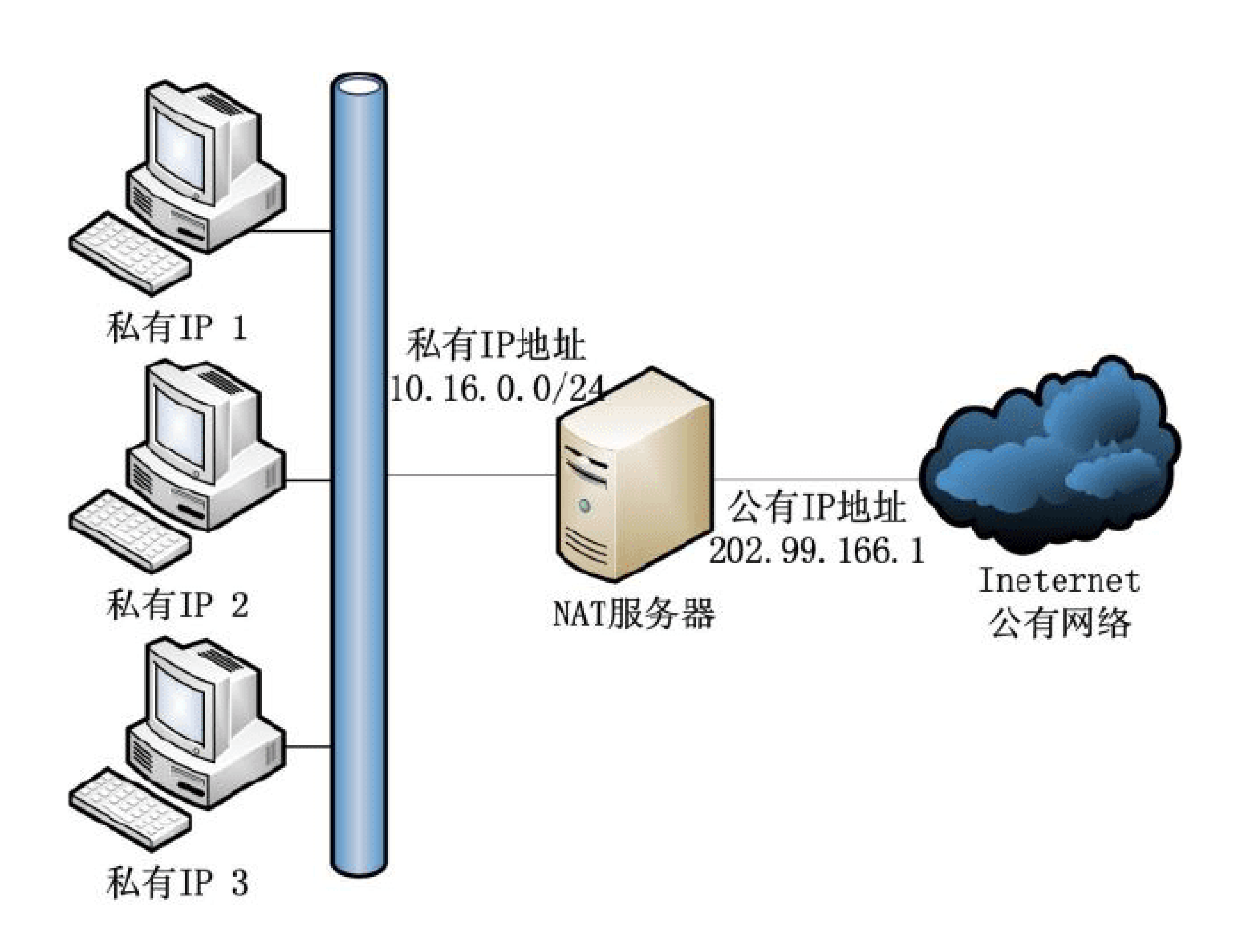


图 11-9 源 NAT 配置案例

以上命令中，“-t nat”指定使用的是 nat 表，“-A POSTROUTING”表示在 POSTROUTING 链中添加规则，“--to-source 202.99.166.1”表示把数据包的源 IP 地址转换为 202.99.166.1，而根据-s 选项的内容，匹配的数据包其源 IP 地址应该是属于 10.16.0.0/24 子网的。还有，“-o eth0”指定了只有从 eth0 接口出去的数据包才做源 NAT 转换，因为从其他接口出去的数据包可能不是到 Internet 的，不需要进行地址转换。

以上命令中，转换后的公网地址直接是 eth0 的公网 IP 地址。也可以使用其他地址，例如，202.99.166.2。此时，需要为 eth0 创建一个子接口，并把 IP 地址设置为 202.99.166.2，使用的命令如下所示。

```
# ifconfig eth0:1 202.99.166.2 netmask 255.255.255.240
```

以上命令使 eth0 接口拥有两个公网 IP。也可以使用某一 IP 地址范围作为转换后的公网地址，此时要创建多个子接口，并对应每一个公网地址。而“--to-source”选项后的参数应该以“a.b.c.x-a.b.c.y”的形式出现。

前面介绍的是数据包转换后的公网 IP 是固定的情况。如果公网 IP 地址是从 ISP 服务商那里通过拨号动态获得的，则每一次拨号所得到的地址是不同的，并且网络接口也是在拨号后才产生的。在这种情况下，前面命令中的“--to-source”选项将无法使用。为了解决这个问题，Iptables 提供了另一种称为 IP 伪装的源 NAT，其实现方法是采用“-j MASQUERADE”目标动作，具体命令如下所示。

```
# iptables -t nat -A POSTROUTING -s 10.16.0.0/24 -o ppp0 -j MASQUERADE
```

以上命令中，ppp0 是拨号成功后产生的虚拟接口，其 IP 地址是从 ISP 服务商那里获得的公网 IP。“-j MASQUERADE”表示把数据包的源 IP 地址改为 ppp0 接口的 IP 地址。

注意： 除了上面的源 NAT 配置外，在实际应用中，还需要配置其他一些有关 Iptables 网络防火墙的规则，同时，路由的配置也是必不可少的。

11.6.6　目的 NAT 配置案例

目的 NAT 改变的是数据包的目的 IP 地址，当来自 Internet 的数据包访问 NAT 服务器网络接口的公网 IP 时，NAT 服务器会把这些数据包的目的地址转换为某一对应的内网 IP，再路由给内网计算机。这样，使用内网 IP 地址的服务器也可以为 Internet 上的计算机提供网络服务了。

位于子网 10.16.0.0/24 的是普通的客户机，它们使用源 NAT 访问 Internet。而子网 10.16.2.0/24 是服务器网段，里面的计算机运行着各种网络服务，它们不仅要为内网提供服务，而且要为 Internet 上的计算机提供服务。但由于使用的是内网地址，因此需要在 NAT 服务器配置目的 NAT，才能让来自 Internet 的数据包能顺利到达服务器网段，如图 11-10 所示。

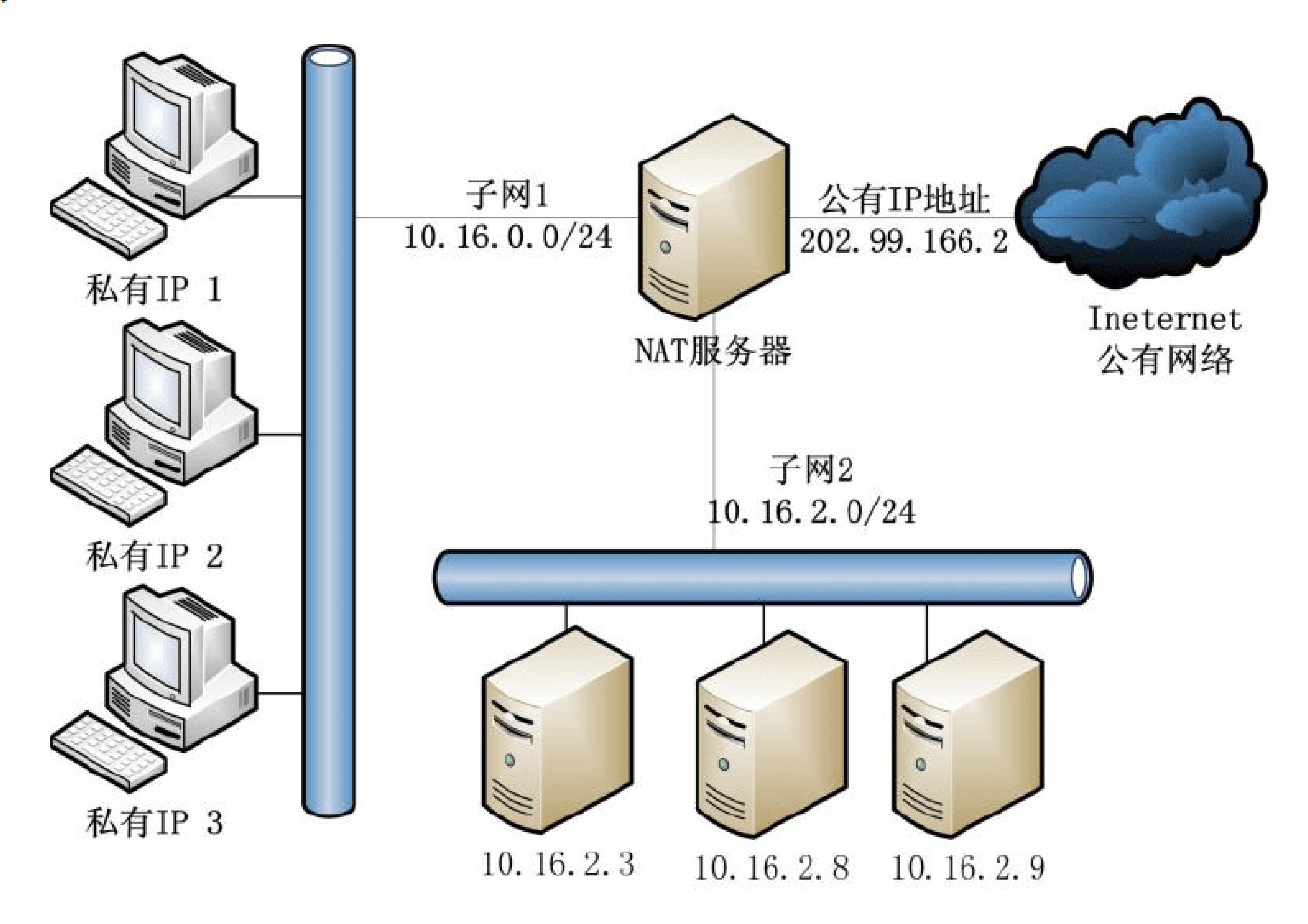


图 11-10　目的 NAT 案例

假设 IP 为 10.16.2.3 的计算机需要为 Internet 提供网络服务，此时，可以规定一个公网 IP 地址，使其与 10.16.2.3 建立映射关系。假设使用的公网 IP 是 202.99.166.2，则配置

目的 NAT 的命令如下：

```
# iptables -t nat -A PREROUTING -i eth0 -d 202.99.166.2/32 -j DNAT --to
10.16.2.3
```

以上命令是在 PREROUTING 链中添加规则，这条链位于路由模块的前面，因此是在路由前改变了数据包的目的 IP，这将对路由的结果造成影响。由于网络接口 eth0 与 Internet 连接，因此，“-i eth0”保证了数据包是来自 Internet 的数据包。“-d 202.99.166.2/32”表示数据包的目的地是 202.99.166.2 主机，而这个 IP 应该是 eth0 某个子接口的地址，这样才能由 NAT 服务器接收数据包，否则，数据包将会因为无人接收而丢弃。

“-j DNAT”指定了目标动作是 DNAT，表示要对数据包的目的 IP 进行修改，它的子选项“--to 10.16.2.3”表示修改后的 IP 地址是 10.16.2.3。于是，目的 IP 修改后，接下来将由路由模块把数据包路由给 10.16.2.3 服务器。

以上是让一个公网 IP 完全映射到内网的某个 IP 上，此时同 10.16.2.3 主机直接位于 Internet，并且使用 202.99.166.2 地址是没有区别的。因此这种方式虽然达到了地址转换的目的，但实际上并没有带来多大好处，因为使用 NAT 的主要目的是为了能够共用公网 IP 地址，以节省日益紧张的 IP 地址资源。为了达到共用 IP 地址的目的，可以使用端口映射。

端口映射是把一个公网 IP 地址的某一端口映射到内网某一 IP 地址的某一端口上去。它使用起来非常灵活，两个映射的端口其端口号可以不一样，而且同一个公网 IP 的不同端口可以映射到不同的内网 IP 地址上去。

例如，假设主机 10.16.2.3 只为外网提供 Web 服务，因此，只需要开放 80 端口，而主机 10.16.2.9 为外网提供了 FTP 服务，因此需要开放 21 号端口。在这种情况下，完全可以把公网 IP 地址 202.99.166.2 的 80 号和 21 号端口分别映射到 10.16.2.3 和 10.16.2.9 的 80 号和 21 号端口，以便两台内网服务器可以共用一个公网 IP。具体命令如下所示。

```
# iptables -t nat -A PREROUTING -i eth0 -d 202.99.166.2/32 -p tcp --
dport 80 -j DNAT --to 10.16.2.3:80
# iptables -t nat -A PREROUTING -i eth0 -d 202.99.166.2/32 -p tcp --
dport 21 -j DNAT --to 10.16.2.9:21
```

以上命令中，目的地址是 202.99.166.2 的 TCP 数据包。当目的端口是 80 时，将转发给 10.16.2.3 主机的 80 端口；当目的端口是 21 时，将转发给 10.16.2.9 主机的 21 号端口。当然，两个映射的端口完全可以不一样。例如，如果还有一台主机 10.16.2.8 也通过 80 端口提供 Web 服务，并且映射的 IP 地址也是 202.99.166.2，此时需要把 202.99.166.2 的另一个端口，如 8080，映射到 10.16.2.8 的 80 端口，命令如下：

```
# iptables -t nat -A PREROUTING -i eth0 -d 202.99.166.2/32 -p tcp --
dport 8080 -j DNAT --to 10.16.2.8:80
```

注意：上面介绍的只是有关 Iptables 中的 DNAT 配置，在实际应用中，还需要其他一些配置的配合才能真正成功。例如，filter 表的 3 个链应该允许相应的数据包通过，应该为每一个外网 IP 创建 eth0 接口的子接口等。

此外，对于 FTP 服务来说，由于 21 号端口只是建立控制连接时用到的端口，真正传输数据时要使用其他端口。而且在被动方式下，客户端向 FTP 服务器发起连接的端口号

是随机的，因此，无法通过开放固定的端口来满足要求。为了解决这个问题，可以在 Linux 系统中载入以下两个模块。

```
modprobe ip_conntrack_ftp
modprobe ip_nat_ftp
```

这两个模块可以监控 FTP 控制流，以便能事先知道将要建立的 FTP 数据连接所使用的端口，从而可以允许相应的数据包通过，即使防火墙没有开放这个端口。

11.7　本 章 小 结

本章对 Linux 服务器安全进行了介绍，重点对 Iptables 进行了探讨。主要介绍了 Iptables 的安装与配置文件、Iptables 规则与 Iptables 的使用方法。本章最后介绍了使用 Iptables 实现 NAT 的方法，并给出了具体的配置案例，方便读者学习。

11.8　课 后 练 习

1. 选择题

(1) 在 Linux 2.4 以后的内核中，提供 TCP/IP 包过滤功能的软件叫(　　)。

A. rarp　　B. route　　C. iptalbles　　D. filter

(2) 在 Linux 操作系统中，可以通过 iptables 命令来配置内核中集成的防火墙，若在配置脚本中添加 Iptables 命令#iptables -t nat -A PREROUTING -p tcp -s 0/0 -d 61.129.3.88 --dport 80 -j DNAT -to-destination 192.168.0.18，其作用是(　　)。

A. 将对 192.168.0.18 的 80 端口转发到内网的 61.129.3.88 主机上
B. 将对 61.129.3.88 的 80 端口转发到内网的 192.168.0.18 主机上
C. 将对 192.168.0.18 的 80 映射到内网的 61.129.3.88 的 80 端口
D. 禁止对 61.129.3.88 的 80 端口访问

(3) 从下面选择关于 IP 伪装的适当描述(　　)。

A. 它是一个转化包的数据工具
B. 它的功能就像 NAT 系统，转换内部 IP 地址到外部 IP 地址。
C. 它是一个自动分配 IP 地址的程序
D. 它是一个连接内部网到 Internet 的工具

(4) 不属于 Iptables 操作的是(　　)。

A. ACCEPT　　B. DROP 或 REJECT
C. LOG　　D. KILL

(5) 假设要控制来自 IP 地址 199.88.77.66 的 ping 命令，可用的 Iptables 命令(　　)。

A. iptables -a INPUT -s 199.88.77.66 -p icmp -j DROP
B. iptables -A INPUT -s 199.88.77.66 -p icmp -j DROP
C. iptables -A input -s 199.88.77.66 -p icmp -j DROP
D. iptables -A input -S 199.88.77.66 -p icmp -j DROP

(6) 如果想要防止 199.88.77.0/24 网络用 TCP 分组连接端口 21，Iptables 命令是(　　)。

A. iptables -A FORWARD -s 199.88.77.0/24 -p tcp --dport 21 -j REJECT

B. iptables -A FORWARD -s 199.88.77.0/24 -p tcp --dport 21 -j REJECT C

C. iptables -a forward -s 199.88.77.0/24 -p tcp --dport 21 -j reject

D. iptables -A FORWARD -s 199.88.77.0/24 -p tcp --dport 21 -j DROP

(7) 在 Iptables 中，特殊目标规则 REJECT 表示(　　)。

A. 让数据包透明通过

B. 简单地丢弃数据包

C. 丢弃该数据，同时通知数据的发送者数据被拒绝通过

D. 被伪装成是从本地主机发出的，回应的数据被自动地在转发时解伪装

(8) 一台 CentOS 5 主机配置了防火墙，若要禁止客户机 192.168.1.20/24 访问该主机的 Telnet 服务，可以添加(　　)规则。

A. iptables -A INPUT -p tcp -s 192.168.1.20 --dport 23 -j REJECT

B. iptables -A INPUT -p tcp -d 192.168.1.20 --sport 23 -j REJECT

C. iptables -A OUTPUT -p tcp -s 192.168.1.20 --dport 23 -j REJECT

D. iptables -A OUTPUT -p tcp -d 192.168.1.20 --sport 23 -j REJECT

(9) 在配置 netfilter/iptables 时，通常需要开启设备的转发功能，下列(　　)可以完成转发功能的开启。

A. 给该设备设置正确的网关地址

B. vi /proc/sys/net/ipv4/ip_forward　将该文件值设置为“1”

C. echo　“1”　>/proc/sys/net/ipv4/ip_forward

D. vi /etc/sysconf/network　将 NETWORKING=YES

2. 判断题

Linux 内核通过 netfilter/iptables 实现的防火墙功能属于包过滤防火墙，可以实现以下功能：作为主机防火墙实现外部网络与主机之间的访问控制；作为网络防火墙实现外部网络与内部网络的访问控制；作为网关服务器实现网络地址转换(NAT)功能，实现内部网络通过网关主机共享访问外部网络。　(　　)

3. 简答题

(1) 简述防火墙的概念、作用。

(2) 简述 Iptables 的工作过程。

(3) 简述 NAT 的工作过程。

附录

Linux 常用词汇及术语大全

对于初涉 Linux 世界的用户而言，有许多新的术语需要学习。本附录的词汇表简明地解释了 Linux 产品常用的许多术语、首字母缩写词和缩写的意思和意义(其中有些术语并不是 Linux 所特有的，初学用户可能仍然对它们感到很陌生)。本文档将有助于澄清对 Linux 的一些混淆；但是，这里并没有包含最常用的硬件、软件和通信方面的术语。

术 语	对应英文名	含 义
帐户名称	Account Name	等同于登录标识、用户标识或用户名。是指派给 UNIX/Linux 系统上用户的名称。可以在系统上对多个用户设置唯一的账户名称，每个用户具有不同的访问(权限)级别。在安装完 Linux 之后，账户名称由超级用户(Superuser)或 root 操作员指派
AfterStep	AfterStep	用户界面(窗口管理器)之一，AfterStep 使得 Linux 的外观很像 NeXTSTEP，而且还有些增强功能。要获取更多关于 AfterStep 的信息，请访问 www.afterstep.org(另请参阅"Enlightenment"、"GNOME"、"KDE"和"X Window 系统")
Awk	Aho、Weinberger 和 Kernighan	一种编程语言，因其模式匹配语法而特别有用，通常用于数据检索和数据转换。一个 GNU 版本称为 Gawk
高级电源管理	APM(Advanced Power Management)	一种工业标准，它允许系统处理器和各个组件进入省电模式，包括挂起、睡眠和关机。APM 软件对于移动设备尤为重要，因为它节省了电池电量
附加符号	Append Symbol	两个键盘字符 >(也就是 >>)。通常用它将命令的输出发送到文本文件，将数据附加到文件的尾部，而不是替换现有的内容。例如，ls -a >> output.txt 将当前目录列表发送到名为 output.txt 的文件，并将其添加到该文件的尾部。重复执行该命令会不断地将新数据添加到文件尾部(另请参阅"管道符号"和"重定向符号")
归档文件	Archive	含有多个文件的单个大型文件，通常对其进行压缩以节省存储空间。经常创建归档文件以方便计算机之间的传送。流行的归档格式包括 ARJ、TAR、ZIP 和 ZOO。它们都可以用来创建这样的归档文件
后台进程	Background Process	运行时无须用户输入的程序。可以在诸如 UNIX/Linux 之类的多任务操作系统上运行多个后台进程，而用户则与前台进程交互(例如，数据输入)。有些后台进程(例如，守护程序)从来都不需要用户输入。其他一些进程只是在用户忙于目前运行于前台的程序时才临时处于后台
Bin	Bin	一个含有可执行程序的目录，这些程序主要是二进制文件
引导盘	Boot Disk	一张软盘，其中含有操作系统(如 Linux)引导(启动)计算机并从命令行运行一些基本程序所需的足够内容。如果因某种原因导致系统表现为无法引导，那么引导盘是必需的。引导盘还用于对硬盘进行分区和格式化、恢复主引导记录(Master Boot Record)或者复制特定文件等

续表

术 语	对应英文名	含 义
BSD Unix	BSD Unix	加州大学伯克利分校开发的 UNIX
公共网关接口	CGI	在 Web 服务器上，用来在脚本和 / 或应用程序之间传输数据，然后将该数据返回给 Web 页面或浏览器。CGI 脚本经常是使用 Perl 语言创建的，它能够生成动态 Web 内容(包括电子商业购物篮、讨论组、调查表单以及实时新闻等)
编译器	Compiler	用于将编程源代码转换成可执行程序的程序
守护程序	Daemon	操作系统的后台进程，通常具有 root 安全级别许可权。守护程序通常隐藏在后台，直至被某个事件(例如特定的时间或日期、时间间隔、收到电子邮件等)触发后它才会进入活动状态
桌面	Desktop	操作系统用户界面，旨在表示一个在上面放东西的办公桌。操作系统的桌面并不使用有形的电话、电灯、收 / 发箱等，而是使用程序及数据图标、窗口、任务栏和类似的东西。Linux 可以使用许多不同的桌面环境，包括 KDE、GNOME 和 X11，它们可以由用户安装(另请参阅“GUI”、“窗口管理器”和“X Window 系统”)
文件系统	FileSystem	一组程序，它们告诉操作系统如何访问及解释存储在磁盘或磁带驱动器或者其他存储媒介上的内容。常见的文件系统包括：FAT 和 FAT-32(DOS/Windows)、HPFS(OS/2)、NFS、NTFS(Windows NT/2000)以及其他文件系统
过滤器	Filter	一种程序，它(从文件、程序输出或命令行输入)读取数据作为输入，根据一组预定义条件处理输入(如，按字母顺序排序)，然后输出处理过的数据。一些常见的过滤器包括 Awk、Grep、Sed 和 Sort
前台进程	ForegroundProcess	在多任务操作系统(诸如 UNIX/Linux)中，前台进程是用户当前与之交互的程序(例如，数据输入)。随着用户在程序之间切换，会导致这些程序在不同的时刻处于前台。在层叠的窗口环境中，前台进程是最前面的窗口
GNU C 编译器	GNU C Compiler 或者 GCC	由 GPL 管理的一个高质量 C 编译器
GNOME 桌面环境	GNU Network Object Model Environment	一种用于 Linux 的用户界面(窗口管理器)，它是用 Gtk 构建的。更多关于 GNOME 的信息，请访问 www.gnome.org
图形用户界面	Graphical User Interface	图标、窗口及屏幕上其他图形图像的集合，它们提供了用户与操作系统交互的方法(另请参阅“桌面”和“窗口管理器”)
Home 目录		用户登录之后所在的目录

续表

术　语	对应英文名	含　义
超文本标记语言	Hyper Text Markup Language	用于设计 Web 页面的标准标记语言。标记“tag”或格式化命令允许 Web 页面设计人员确定突出显示、定位图形及创建超链接等
超文本传输协议	Hyper Text Transport Protocol(HTTP)	一组创建的准则，用于请求和发送基于 HTML 的 Web 页面
Init	Init	操作系统装入后立即运行的第一个进程。它以单用户方式启动系统或生成 shell 来读取启动文件，并打开指定用于登录的端口
解释型语言	Interpreted Language	与编译型程序不同，每次运行解释型程序时都要由解释器程序实时地将源代码转换成二进制形式，而编译型程序由编译器一次性将源代码转换成可执行代码，随后从其二进制形式运行。解释型语言(以及用它们编写的程序)往往要比编译型语言及伪代码语言 / 程序慢，并且通常只有有限的底层操作系统功能访问权限或直接访问硬件的权限。但从另一角度来说，它们无需编译器(可能非常昂贵)，并且经常包含在操作系统中，通常比编译型语言更容易编程。解释型语言的例子有 BASIC、Perl、Python 和 REXX/Object REXX
Java	Java	Sun Microsystems 开发的、独立于操作系统的面向对象编程语言。Java 通常用于 Web 服务器。Java 应用程序和 applet 有时以下载的形式提供给用户，以便在他们的系统上运行。Java 编程语言可以编制应用程序或较小的 Java“applet”。Java 是 C++ 语言稍加简化的版本，通常是进行解释而不是编译
Java 开发工具箱	Java Development Kit(JDK)	由 Sun、IBM 或其他公司开发的 Java 编程工具箱，可以用于 UNIX/Linux 及其他操作系统
面向对象	Object-Oriented	一种软件开发方法，它为程序员提供标准可重用的软件模块(组件)，而无需开发人员每次都编写定制编程代码。使用标准组件缩短了开发时间(因为其他程序员已经编写并测试了这些组件)，并且通过使用相同的组件确保了程序具有标准的外观
开放源码	Open Source	一个稍显模糊的术语，是指同源代码一起发布的软件。提供源代码这一事实并不一定意味着用户可以修改和重新分发源代码。这个术语有时可以和“免费软件”互换使用，尽管它们的意思并不总是相同
开放源码软件	Open Source Software(OSS)	参阅“开放源码”

续表

术　语	对应英文名	含　义
所有者	Owner	对文件具有访问特权的用户，通常是创建该文件的用户
可插入的认证模块	Pluggable Authentication Modules(PAM)	用于系统安全性的可替换的用户认证模块，它允许在不知道将使用何种认证方案的情况下进行编程。这允许将来用其他模块来替换某个模块，却无需重写软件
面板	Panel	Linux 中对应于 Windows 任务栏的名称
分区	Partition	磁盘驱动器的一个连续部分，它被操作系统当作物理驱动器。这样，可以为一个磁盘驱动器赋予几个驱动器符号
实用摘录与报告语言	Practical Extraction and Report Language(Perl)	一种常用的脚本编制 / 编程语言。经常用在 UNIX/Linux Web 服务器上生成 CGI 脚本
权限	Permission	读写文件和目录及执行程序的权限。超级用户或 root 操作员可以逐个文件、逐个目录地，或者按照账户名称(用户标识)赋予各种权限级别
移植	Port/Ported/Porting	一个过程，即获取为某个操作系统平台编写的程序，并对其进行修改使之能在另一 OS 上运行，并且具有类似的功能。通常很少或者干脆就不尝试定制程序以利用新操作系统的特有能力，这与为某个特定操作系统优化应用程序不同
可移植	Portable	描述一类软件的术语，这类软件旨在只需少量修改和重新编译就可在多个操作系统上使用
进程	Process	正在执行的程序
公共域	Public Domain	可供任何人以任何目的使用和修改的软件，甚至可以将其并入商业软件的分发。公共域软件不保留版权，作者也不保留任何权利
公钥加密	Public Key Encryption	一种包括两个单独密钥(公钥和私钥)的数据加密方法。使用公钥加密的数据只能用私钥解密，反之亦然。一般而言，公钥是公开的，可以用来加密发送给私钥持有者的数据，私钥用来对数据进行签名
Python	Python	一种面向对象伪代码编程语言
独立 / 廉价磁盘 / 设备冗余阵列	Redundant Array of Independent/Inexpensive Disks/Devices(RAID)	一种提供数据冗余、改善性能和 / 或从磁盘崩溃中迅速恢复数据的方法，它是通过在多个磁盘驱动器上分布或复制数据来实现这一点的。常用的 RAID 类型包括 RAID 0(数据条带化)，RAID 1(磁盘镜像)和 RAID 5(具有分布式奇偶校验的条带化)。RAID 配置通常需要 SCSI 磁盘驱动器(而不是 IDE/EIDE)，可能要求磁盘相同(相同的容量、品牌等)。操作系统将 RAID 阵列看作单个设备
Root 操作员		具有执行所有系统级任务权限的用户标识

续表

术　语	对应英文名	含　义
RPM 软件包管理器	RPM Package Manager	一种用于因特网下载包的打包及安装工具，它包含在某些 Linux 分发版中。它生成具有 .RPM 扩展名的文件。与 Dpkg 类似
脚本	Script	一组存储在文件中的命令。用于进行自动重复的执行
会话	Session	用户在登录到注销期间与操作系统之间的完整交互过程
源代码	Source Code	程序员输入的、原始状态的编程命令。有些编程语言允许命令实时地由程序解释器执行。其他语言则要求必须先将命令编译成可执行程序(二进制)后才能使用这些命令。在 UNIX/Linux 世界中，有些软件仅以源代码形式分发；另一些软件包则同时包含源代码和二进制代码；还有一些则仅以二进制格式分发
交换	Swap	暂时将数据(程序和 / 或数据文件)从随机存取存储器移到磁盘存储器(换出)，或反方向移动(换入)，以允许处理比物理内存所能容纳的更多的程序和数据。也称为虚拟内存
同步	Sync	将所有暂挂的输入 / 输出强制写回磁盘驱动器
文本编辑器	Text Editor	用于编辑文本文件的程序。类似于字处理程序，但没有大多数 / 全部格式化功能(例如设置页边距、斜体和字体等)。经常用于书写或编辑脚本、程序和 ASCII 文本文件(如 README.1ST)
线程	Thread	一小段程序，其行为就像是较大程序的一个独立子集，也称为“进程”。多线程程序能够比单个程序或单线程程序运行得快得多，因为它可以并行(而不是串行(顺序))地执行几个甚至多个不同的任务。而且，单个应用程序内的多个线程可以共享资源，并且相互之间可以来回传递数据
TrueType 字体	TrueType	与 PostScript 字体不一样，它们旨在成为与打印机无关的各种字体，可用于 Apple Macintosh 和 Windows，不常用于 UNIX/Linux

课后习题答案

第 1 章

1. 填空题

(1) 开源的，Unix，Unix，硬件平台移植
(2) 图形用户界面

2. 选择题

(1) CD
(2) D
(3) C

3. 判断题

(1) 对　(2) 错

4. 简答题

(1) Linux 操作系统的内核版本指的是在 Linux 本人领导下的开发小组开发出的系统内核的版本号。自 1994 年 3 月 14 日发布了第一个正式版本 Linux 1.0 以来，每隔一段时间就有新的版本或其修订版公布。

不同的系统厂商在开发 Linux 时，虽然可能使用同一个 Linux 内核，但为了确立自己的品牌，都会使用不同的名称为这些发布的版本命名，这就是 Linux 的发行版本。

(2) 请参考 1.2 节内容。

第 2 章

1. 填空题

(1) 从光盘安装，从硬盘安装，网络安装，kickstart 安装，PXE 安装
(2) 文件

2. 选择题

(1) D
(2) D
(3) C

3. 判断题

(1) 对　(2) 错

4. 简答题

(1) 请参考 2.1.3 节内容。
(2) 请参考 2.3.2 节内容。
(3) 请参考 2.4.3 节中的内容，进行自定义面板的操作步骤即可。

第 3 章

1. 填空题

(1) alias tdir="ls -art"，bashrc，bash_profile
(2) ln，硬链接，符号链接
(3) cat >abc，Ctrl+d，Ctrl+c，cp /dev/tty1 abc

2. 选择题

(1) C
(2) A
(3) D
(4) D

3. 判断题

错

4. 简答题

(1) 请参考 3.1.1 节内容。
(2) 请参考 3.3 节内容。

第 4 章

1. 填空题

(1) 动态主机配置协议
(2) IP 地址
(3) 授权
(4) MAC 地址

2. 选择题

(1) ABCD
(2) C
(3) D
(4) B
(5) B

(6) C
(7) B
(8) A
(9) ABCD

3. 简答题

(1) DHCP 的工作过程如下：

① DHCP 客户向 DHCP 服务发出请求，要求租借一个 IP 地址。但由于此时 DHCP 客户上 TCP/IP 还没有初始化，它还没有一个 IP 地址，因此只能使用广播手段向网上所有 DHCP 服务器发出请求。

② 网上所有接收到该请求的 DHCP 服务器，首先检查自己的地址池中是否还有空余的 IP 地址，如果有的话将向该客户发送一个可提供 IP 地址(offer)的信息。

③ DHCP 客户一旦接收到来自某一个 DHCP 服务器的(offer)信息时，它就向网上所有的 DHCP 服务器发送广播，表示自己已经选择了一个 IP 地址。

④ 被选中的 DHCP 服务器向 DHCP 客户发送一个确认信息，而其他的 DHCP 服务器则收回它们的(offer)信息。

(2) DHCP 的优缺点：

在网络中应用 DHCP 有以下优点：减少错误，通过配置 DHCP，把手工配置 IP 地址所导致的错误减少到最低程度，例如已分配的 IP 再次分配给另一设备所造成的地址冲突等将大大减少；减少网络管理，TCP/IP 配置是集中化和自动完成的，不需要网络管理员手工配置；网管员能集中定义全局和特定子网的 TCP/IP 配置信息，使用 DHCP 选项可以自动给客户机分配全部范围的附加 TCP/IP 配置值。

DHCP 缺点：DHCP 不能发现网络上非 DHCP 客户机已经在使用的 IP 地址；当网络上存在多个 DHCP 服务器时，一个 DHCP 服务器不能查出已被其他服务器租出去的 IP 地址；DHCP 服务器不能跨路由器与客户机通信。

(3)

① 181 个

② 该 DHCP 服务器指定的默认网关是：192.168.1.254。

域名是：abc.com。

指定的 DNS 服务器是：192.168.1.1 和 192.168.1.2。

③ 为网卡的 MAC 地址“00:A0:78:8E:9E:AA”的客户端主机分配固定的 IP 地址：192.168.1.22。

为该客户分配主机域名：fixed.abc.com。

④ 通过 ipconfig/all 命令可以得到客户端 TCP/IP 当前的详细配置信息。

第 5 章

1. 填空题

(1) network file system，网络文件系统

(2) portmap，rpc.nfsd，rpc.mountd

(3) Linux，Unix

(4) showmount –a 192.168.1.103

(5) chkconfig –level 345 portmap on，chkconfig –level 345 nfs on

2. 选择题

(1) C

(2) C

(3) D

(4) A

(5) D

(6) B

(7) A

(8) C

(9) C

(10) B

3. 简答题

(1) NFS 即网络文件系统(Network File System)，是使不同的计算机之间能通过网络进行文件共享的一种网络协议，主要用于 UNIX/Linux 操作系统中。使用 mount 命令或者 mount –t nfs 对 NFS 进行挂载。

(2)

① rpc.nfsd：它是基本的 NFS 守护进程，主要功能是管理客户端是否能够登入服务器。

② rpc.mountd：它是 RPC 安装守护进程，主要功能是管理 NFS 的文件系统。当客户端顺利地通过 rpc.nfsd 登录 NFS 服务器后，rpc.mountd 会读取 NFS 的配置文件/etc/exports 来对比客户端的权限。

③ portmap: portmap 的主要功能是进行端口映射工作。当客户端尝试连接并使用 RPC 服务器提供的服务(如 NFS 服务)时，portmap 会将所管理的与服务对应的端口号提供给客户端，从而使客户端可以通过该端口向服务器请求服务。

第 6 章

1. 填空题

(1) 域名与 IP 地址

(2) ps -ef | grep named

(3) 正向区域数据库，反向区域数据库，根域数据库文件

2. 选择题

(1) C
(2) A
(3) C
(4) B
(5) D
(6) BC
(7) ABD
(8) BC

3. 简答题

(1) 用户要想通过使用 InternetExplorer 来访问万维网服务器，则用户必须首先获的与万维网服务器的正式域名相关的 IP 地址，依靠 DNS 及 WINS 将主机名称转换为 IP 地址，这个过程被称为主机名称解析(NameResolution)。一旦用户的计算机将 WWW 服务器的正式名称解析为它的 IP 地址，它就可以与 WWW 服务器建立起 TCP/IP 网络通信。

(2) 将主机名称转换为 IP 地址，这个过程被称为主机名称解析(域名解析)。

(3) 当局部 DNS 服务器自己不能回答客户机的 DNS 查询时，它就需要向其他 DNS 服务器进行查询。局部 DNS 服务器自己负责向其他 DNS 服务器进行查询，一般是先向该域名的根域服务器查询，再由根域名服务器一级级向下查询。最后得到的查询结果返回给局部 DNS 服务器，再由局部 DNS 服务器返回给客户端。

第 7 章

1. 填空题

(1) /etc/samba/smb.conf
(2) system-config-samba
(3) 该用户在系统中已经存在
(4) Server Message Block

2. 选择题

(1) B
(2) B
(3) D
(4) ACD
(5) AB

3. 简答题

(1) Samba 服务器的配置步骤：

① 检查 Samba 服务软件包是否安装，入未安装则需要先安装 samba 服务软件包。

② 根据需要创建共享目录。
③ 修改 smb.conf 主配置文件中相应参数。
④ 创建 Samba 访问用户，如服务器只开启匿名访问，则此步骤跳过。
⑤ 启动 Samba 服务器。
⑥ 编辑防火墙规则或者关闭防火墙。并使 SELinux 允许 Samba 服务运行。
⑦ 在客户端对 Samba 服务器进行测试。

(2) 可实现在 Linux、Unix、Windows 之间共享文件。

第 8 章

1. 填空题

(1) 通用资源标志符 Uniform Resource Identifier

(2) 静态 Web 服务，动态 Web 服务

2. 选择题

(1) A

(2) D

3. 简答题

(1) 要启动 WWW 服务，需要执行命令：

```
# /etc/rc.d/init.d/httpd -k start
要停止 www 服务，应输入命令：
# /etc/rc.d/init.d/httpd -k stop
要重新启动 www 服务，应输入命令：
# /etc/rc.d/init.d/httpd -k restart
要查看 www 服务是否运行，应输入命令：
# ps aux | grep httpd
```

(2) Apache 支持两种类型的虚拟主机.基于 IP 的虚拟主机和基于名字的虚拟主机。基于 IP 的虚拟主机要求有多个合法的 IP 地址，而基于名字的虚拟主机则不受 IP 地址的限制，允许用户创建无限多个虚拟主机。

4. 操作题

(1) 请参考 8.7.2 节的内容，在 httpd.conf 中配置目录访问权限。

(2) 请参考 8.6.2 节的内容，在 httpd.conf 中分别为两个 IP 地址配置虚拟主机。

(3) 请参考 8.8.1 节的内容，按照讲述的操作步骤安装即可。

(4) 请参考 8.8.2 节的内容，按照讲述的操作步骤安装即可。

(5) 请参考 8.8.3 节的内容，按照讲述的操作步骤安装即可。

第 9 章

1. 填空题

(1) File Transfer Protocol，文件传输协议
(2) POST(主动)模式，RASV(也称被动)模式

2. 选择题

(1) D
(2) B
(3) B

3. 简答题

(1) 在安装操作系统时，如果选择作为服务器，系统将自动安装 VsFTPd FTP 程序。要使 ftp 服务能够启动，单击 “主菜单”→“系统设置”→“服务器设置”→“服务”，或在终端方式下输入命令“redhat-config-services”，屏幕上将出现“服务配置工具”对话框，找到并启动 VsFTPd 服务。

VsFTPd 的主要配置文件是/etc/vsftpd/vsftpd.conf。

(2) 是在 /etc/services 里面设置的，此外，正规的 ports 在 command 是 21 而 data 是 20。

第 10 章

1. 填空题

(1) Sendmail，Postfix，Qmail
(2) main.cf，/etc/postfix

2. 选择题

(1) A
(2) ABC

3. 简答题

(1) MUA：邮件用户代理(Mail User Agent)是在邮件使用终端上运行的程序，主要负责编辑和发送邮件，以及从服务器上下载、管理、阅读和处理邮件。目前常用的邮件用户代理在 Windows 平台主要有 Outlook、Foxmail、Dreammail 等；Linux 平台主要有 Evolution、Thunderbird、KMail 等。

MTA：邮件传输代理(Mail Transfer Agent)主要用于存储和发送邮件，也可以说成是邮件服务器软件的总称，如 Sendmail、Postfix、Qmail、Exim 等。一台服务器可以安装多个

MTA，但同一时刻只能有一个 MTA 工作。

(2) 请参考 10.1.2 节的内容。Postfix 服务器与 Sendmail 服务器相比，安全性更强，功能更加完善，同时又兼容使用最广泛的 Sendmail 服务器。ostfix 服务器与 Qmail 服务器相比，功能更加强大，支持软件更加丰富。

4. 操作题

(1)

```
vi /etc/postfix/main.cf
myhostname = t1.ckhitler.org
myorigin = $myhostname
inet_interfaces = all
mydestination = $myhostname,localhost
mynetworks = 127.0.0.0/8, 192.168.6.0/24
relay_domains = $mydestination
alias_maps = hash:/etc/postfix/aliases
alias_database = hash:/etc/postfix/aliases
```

保存退出

```
service postfix restart
```

发送邮件给 root

```
echo "testing…"|mail -s "from ckhitler" root
```

检查邮件

```
Mail
```

(2) 请参考 10.2 节内容。

(3) 请参考 10.3 节内容。

第 11 章

1. 选择题

(1) C

(2) B

(3) B D

(4) D

(5) B

(6) A

(7) C

(8) AD

(9) BC

2. 判断题

对

3. 简答题

(1) 防火墙是建立在内外网络边界上的过滤封锁机制。一般来说，内部网络被认为是安全和可信赖的，而外部网络被认为是不安全和不可信赖的。防火墙的作用是防止不希望的、未经授权的通信进出被保护的网络，迫使一个组织强化自己的网络安全策略，被认为是在可信的内部网络和不安全可信的外部网络之间提供的一个强化内部网络的安全政策。

(2) Iptables 的作用就是添加 Netfilter 提供规则，这些规则告诉 Netfilter 对于从某些地方来、到某些地方去的或者具有某些特定特征的数据包采用什么样的动作。如果一个数据包与一条规则匹配，那么 Netfilter 上的钩子函数就会使用规则所指定的目标(ACCEPT、REJECT、DROP)允许该数据包通过或者阻塞该数据包。

(3) NAT 为网络地址转换。借助 NAT，使用私有 IP 地址的内部网络通过路由器或防火墙向外发送数据时，私有 IP 地址会被转换为合法的公有 IP 地址，而当返回的数据包到达路由器或防火墙时，再将公有 IP 转换为私有 IP 地址。所以一个局域网只需要有少量合法 IP 即可实现多台机器联网的需求。NAT 极大程度上解决了 IPv4 地址匮乏的问题。